Université Paris 13

Doctoral Thesis

Video stabilization : A synopsis of current challenges, methods and performance evaluation

---

Author:
Wilko Guilluy

Supervisors:
Pr Azeddine Beghdadi, Université Paris 13. Directeur de thèse
Dr Laurent Oudre, Université Paris 13. Co-directeur de thèse

Examiners:
Pr Ioan Tabus, Tampere University of Technology. Rapporteur
Pr Titus Zaharia, TELECOM SudParis. Rapporteur
Pr Abdelaziz Bensrhair, INSA Rouen. Examinateur
Pr Anissa Mokraoui, Université Paris 13. Examinatrice

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

L2TI
Ecole doctorale Galilée

December 20, 2018

Abstract

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. 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. However, the ease in accessing visual information through the increasingly easy-to-handle sensors has led to a situation where the number of videos distributed over the Internet is constantly increasing and it becomes difficult to effectively correct all the distortions and artifacts that may result from the signal acquisition. As an example, more than 600000 hours of videos are uploaded each day on Youtube. One of the most perceptually annoying degradation is related to the image instability due to camera movement during the acquisition. 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. There exist some hardware solutions such as tripods, dollies, electronic image stabilizers or gyroscope based technologies that prevent video from blurriness and oscillations. However, their use is still limited to professional applications and as a result, most amateur videos contain unintended camera movements. In this context, the use of software tools, often referred to as Digital Video Stabilization (DVS), seems to be the most promising solution. Digital video stabilization aims at creating a new video showing the same scene but removing all the unintentional components of camera motion. Video stabilization is useful in order to increase the quality and the visual comfort of the viewer, but can also serve as a pre-processing step in many video analysis processes that use object motion, such as background substraction or object tracking.

Video Stabilization (VS) has been an active area of research in the last two decades. More than one hundred methods have been proposed in the literature, and most of these methods are composed of several functional blocks, i.e processing steps such as motion estimation, modeling or removal. This thesis offers 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. This overview is conducted by using an incremental approach based on the essential steps of common video stabilization method. Following this approach, we outline and discuss the different components of digital video stabilization, and provide insights on the current challenges in this hot and evolving topic.

One of the unsolved problem in the field of VS is Video Stabilization Quality Assessment (VSQA). As such, video stabilization evaluation is a multi-criterion problem that could not be easily expressed through some mathematical multi-objective optimization schemes used in computer vision. Indeed, 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 not be mathematically tractable. This is mainly due to the fact that visual discomfort and other perceptual annoying effects are inherently subjective, making objective evaluation rather difficult. This is probably why although many video stabilization methods have been proposed, a little attention has been paid to video stabilization quality assessment. Very often the quality of the processed video is evaluated simply by visual inspection or with basic quantitative measures such as PSNR-based metrics. However, these quantitative measures do not exploit any knowledge nor well-defined model of the visual discomfort due to video instability. Furthermore, 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. This thesis provides a rigorous study and discussion of the existing VSQA metrics and then propose a framework for developing a methodology for effective VSQA. By confronting subjective evaluation experiments and objective measurements, this work is an attempt to lay the cornerstone towards a universal evaluation methodology.

Video stabilization operates in several interdependent steps. In a nutshell, the video motion field is estimated in order to compute the original camera path. The camera path is then smoothed in order to obtain more coherent movements and a new stable and perceptually pleasant video. While each step has its own set of difficulties, the most challenging aspects of video stabilization lie in the estimation of the camera motion and the evaluation of the video stabilization quality. For instance, all detected movements do not convey reliable information about the camera motion. Indeed, movements caused by moving objects are not the result of the camera motion and can lead to errors if not removed. Such challenges are widely studied in the literature. In this thesis, we propose a new method to identify and remove movements caused by objects rather than the camera motion. By recording the displacement in the video of a small set of interest points, we discriminate and identify the object and camera motion using the duration of the trajectory and the characteristics of its movements. This feature

trajectories selection strategy first analyzes each feature trajectory on a local time-window, in order to account for its duration and movement properties. Two local weights are defined that rank each trajectory according to its duration and its adequacy with the movements observed on a time-window centered on a given frame. These local weights are then combined in order to form a global trajectory weight that accounts for the phenomenon observed during the whole duration of the trajectory. Finally, the feature trajectories with the largest weights are selected to estimate the camera motion parameters. Results on a dataset of 15 videos show that this approach outperforms standard outlier removal procedures such as RANSAC.

Résumé en français

Le développement continu de capteurs vidéo et leurs miniaturisations ont étendus leurs usages dans diverses applications allant de la vidéo-surveillance aux systèmes de chirurgie assisté par ordinateur et l'analyse de mouvements physiques et de phénomènes astrophysiques. De nos jours, il est devenu possible de capturer des séquences vidéos dans n'importe quel environnement, sans de lourds et complexes ajustements comme c'était le cas avec les anciens capteurs vidéos. Cependant, l'aisance à accéder à l'information visuelle à travers des capteurs de plus en plus faciles à manipuler a conduit à une situation où le nombre de vidéos distribuées sur internet est en progression constante et il devient difficile de corriger efficacement toutes les déformations et artefacts qui découlent de l'acquisition du signal. Par exemple, plus de 600000 de vidéos sont chargées sur Youtube chaque jour. Une des dégradations les plus gênantes pour la vision humaine est liée à l'instabilité de l'image due aux mouvements de la caméra lors de l'acquisition du signal. Cette source de dégradations se manifeste sous la forme d'oscillations incontrôlées de la trame entière et peut être accompagnée par un effet de flou. Cela affecte la qualité de l'image et produit un inconfort visuel. Il existe des solutions mécaniques telles que les tripodes, chariots, stabilisateurs électroniques ou des technologies s'appuyant sur les gyroscopes qui empêchent les effets de flou ou les oscillations. Cependant, leur utilisation reste limitée à des applications professionnelles et en conséquence, la plupart des vidéos amateurs contiennent des mouvements de caméra non intentionnels. Dans ce contexte, l'utilisation de méthodes numériques, souvent nommées stabilisation de vidéos numérique, semble être une solution prometteuse. La stabilisation numérique cherche à crée une nouvelle vidéo montrant la même scène mais en supprimant toutes les composantes non intentionnels du mouvement de caméra. La stabilisation vidéo est utile pour améliorer la qualité et le confort visuel du spectateur, mais peut aussi servir d'étape de prétraitement pour de nombreux procédés d'analyse vidéo utilisant le mouvement, tel que la soustraction de l'arrière-plan ou le suivi d'objet.

La stabilisation de vidéo a été un thème de recherche active pendant les vingt dernières années. Plus d'une centaine de méthode ont étés proposées dans la littérature, et la plupart de ces méthodes sont composées de plusieurs bloques, par exemple des étapes telles que l'estimation, la modélisation ou la suppression de mouvements. Cette thèse offre une analyse poussée et détaillée des méthodes

proposées en se focalisant sur les approches les plus représentatives développées durant les vingt dernières années. Un des objectifs majeur de cette contribution vise à souligner quelques limitations du processus de stabilisation ainsi que le manque de méthodologie pour comparer le grand nombre de techniques développées. Cette analyse se fonde sur les étapes essentielles des méthodes standards de la stabilisation de vidéos avec une approche incrémentale. Après cette approche nous présentons et commentons les différents composants de la stabilisation numérique, et apportons un aperçu des différents défis dans ce sujet en évolution. Un des aspects non résolu dans ce domaine est l'évaluation de la qualité de la stabilisation. L'évaluation de la stabilisation de vidéo est un problème à multiples critères qui ne peut être facilement exprimé à travers une méthode d'optimisation à critères multiples telle qu'on en utilise en vision par ordinateur En effet, l'inconfort visuel causé par le mouvement de la caméra, les artefacts liés au processus de stabilisation tels que la perte de résolution et les distorsions contribuent tous à la qualité finale de la vidéo de sortie Tous ces artefacts et distorsions ne sont pas aisément formulés mathématiquement Cela vient principalement du fait que l'inconfort visuel et autres effets gênants pour la perception humaine sont intrinsèquement subjectifs, ce qui rend l'évaluation objective difficile. C'est probablement pourquoi de nombreuses méthodes de stabilisations ont été proposées, mais peu d'attention a été prêtée à l'évaluation de la qualité de la stabilisation. Très souvent, la qualité de la vidéo traitée est évaluée par simple inspection visuelle ou par des mesures quantitatives simples comme celles basées sur le PSNR. Cependant ces mesures quantitatives n'exploitent pas de connaissances ou de modèles bien définis de l'inconfort visuel que cause l'instabilité de la caméra. De plus, ces mesures ne prennent en compte que de petits groupes de caractéristiques du mouvement qui sont considérées comme les sources principales de distorsions gênantes. Cette thèse fourni une étude et analyse rigoureuse des méthodes existantes d'évaluation de qualité et propose un cadre pour développer une méthodologie pour une évaluation efficace de la stabilisation. En confrontant expériences d'évaluation subjectives et mesures objectives, ce travail tente de poser les bases d'une future méthode d'évaluation universelle.

La stabilisation de vidéo opère en plusieurs étapes interdépendantes. Simplement, le champ de mouvements est estimé pour obtenir les déplacements de caméra originaux Le chemin de la caméra est ensuite lissé pour obtenir des mouvements plus cohérents et une nouvelle vidéo stable et plaisante visuellement. Si chaque étape a ses propres difficultés, l'aspect le plus difficile est l'estimation du mouvement de la caméra et l'évaluation de qualité de stabilisation. Par exemple, tous les mouvements détectés ne contiennent pas des informations fiables sur le mouvement de caméra. En effet, les déplacements dus à des objets en mouvements ne découlent pas du mouvement de la caméra et peuvent créer des erreurs s'ils ne sont pas détectés et supprimés. Ces défis sont étudiés dans la littérature. Dans cette

thèse, nous proposons une nouvelle méthode pour identifier et supprimer les mouvements causés par des objets mobiles plutôt que le mouvement de caméra. En enregistrant les déplacements dans la vidéo d'un petit groupe de points d'intérêts, nous discriminons et identifions les mouvements d'objets et de caméra en utilisant la durée des trajectoires et les caractéristiques des mouvements. Cette sélection de trajectoires analyse chaque trajectoire dans une fenêtre temporelle locale, afin de tenir compte de ses caractéristiques de mouvement et de durée. Deux poids locaux sont définis pour trier chaque trajectoire selon sa durée et la pertinence de ses mouvements observés dans une fenêtre temporelle centrée sur la trame considérée. Ces poids locaux sont combinés pour former un poids global qui puisse rendre compte des phénomènes observés pendant toute la trajectoire considéré. Enfin, les trajectoires avec les poids les plus importants sont sélectionnés pour estimer les paramètres du mouvement de caméra. Les résultats sur un set de 15 vidéos montre que cette approche est plus performante que des méthode de suppression de valeurs aberrantes tel que RANSAC.

Remerciements

Je souhaite remercier tous ceux et celles qui m'ont aidé et accompagné pendant mes trois années de doctorat et qui ont permis la rédaction de cette thèse. Tout d'abords, je remercie mon directeur de thèse Azeddine Beghdaddi et mon co-directeur Laurent Oudre, pour leurs conseils, leurs contributions scientifiques et pour m'aider à tenir le rythme de travail nécessaire. Je suis particulièrement reconnaissant à Laurent pour son aide lors de ma deuxième année, à Azeddine pour m'avoir soutenu lors des conférences, et à leur soutient lors de la rédaction et la préparation de la soutenance. Je remercie également toute l'équipe du L2TI pour leur accueil chaleureux et leur aide, notamment mes camarades doctorants. Je remercie mes parents, qui m'ont soutenus pendant les moments difficiles, et m'ont aider à préparer mes présentations. Je salut tout particulièrement ma mère, qui m'a laissé vivre chez elle et m'a poussé au travail lorsqu'il le fallait, et a su m'aider pour à travers mes difficultés du monitorat à la soutenance. Merci aussi à ma soeur, qui à pu compatir avec moi des difficultés d'enseignement et de doctorant. Je remercie aussi mes amis creusois, notamment mes camarades de Donjons et Dragons, pour ces parties délirantes et qui m'ont parfois permis de me défouler quand j'en avais besoin.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Chapter 1

Introduction

1.1 Context and motivations

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 ease in accessing visual information through the increasingly easy-to-handle sensors has led to a situation where the number of videos distributed over the Internet is constantly increasing and it becomes difficult to effectively correct all the distortions and artifacts that may result from the signal acquisition.

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. Professional videos are captured using mechanical stabilizers such as tripods or dollies [4] to enforce carefully planned camera movements. There also exist some hardware solutions such as electronic image stabilizers or gyroscope based technologies that prevent video from blurriness and oscillations [5] - see Figure 1.1. 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 [6].

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. This

thesis 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. The process of video stabilization is illustrated on Figure 1.2 and Figure 1.3.

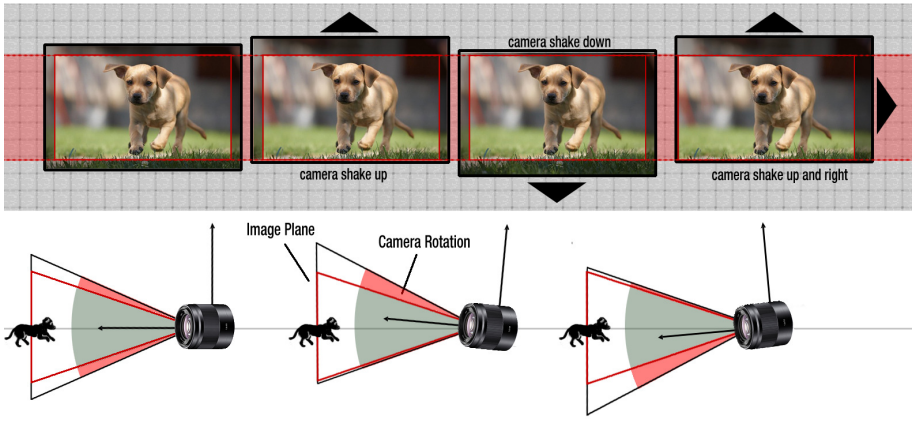

Figure 1.1 *Video stabilization based on additional motion sensors. Electronic Image Stabilization (EIS) is a highly effective method of compensating for hand jitter that manifests itself in distracting video shake during playback. EIS relies on an accurate motion sensor for tracking the source of jitter,which may be hand shake or vehicle motion for example. The motion information is then integrated during the current video frame and used to compensate for it by cropping the viewable image from a stream of video frames thru the imaging pipeline.* Source : TDK InvenSense solutions for video stabilization

These instabilities can produce different types of degradation. Abrupt motion, often seen when using hand-held devices, or high-frequency tremors, such as those felt on a moving vehicle, can cause important visual discomfort [7], [8]. 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 [9]. Finally, for a camera equipped with a rolling shutter sensor, fast camera movements can induce deformations in the scene [10], [11]. Digital stabilization aims at creating a new video showing the same scene but removing all these unintentional components of camera motion.

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 [4]. 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

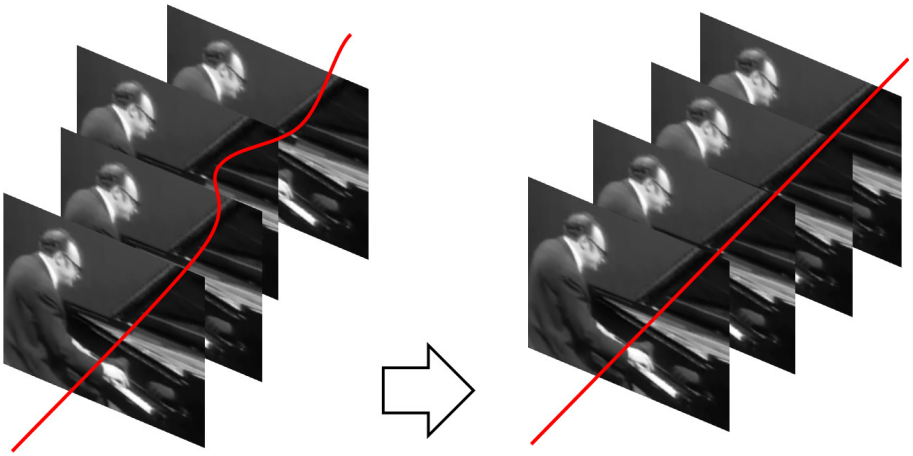

Figure 1.2 Illustration of video stabilization.Video stabilization aims at removing unintentional movements and create a smooth video.

suffer from sever shakes whenever the wearer is running [12]. Video surveillance cameras can also suffer from detrimental jitter, often due to meteorological conditions [13]. Stabilizing such videos can make their exploitation much easier. Other fields that can benefit from this are the medical field with camera-assisted surgery, or remote control of unmanned aerial vehicles [14]. 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 object tracking [13].

While digital stabilization can use additional information from gyroscopes or accelerometers [7], or different viewpoints [15] to improve or facilitate the process, most methods only rely on the video sequence taken from a single camera. Early methods used 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 [16]. 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 computation technologies [17]. 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 attempt to obtain visually-plausible rather than physically-accurate videos, has emerged more recently [18].

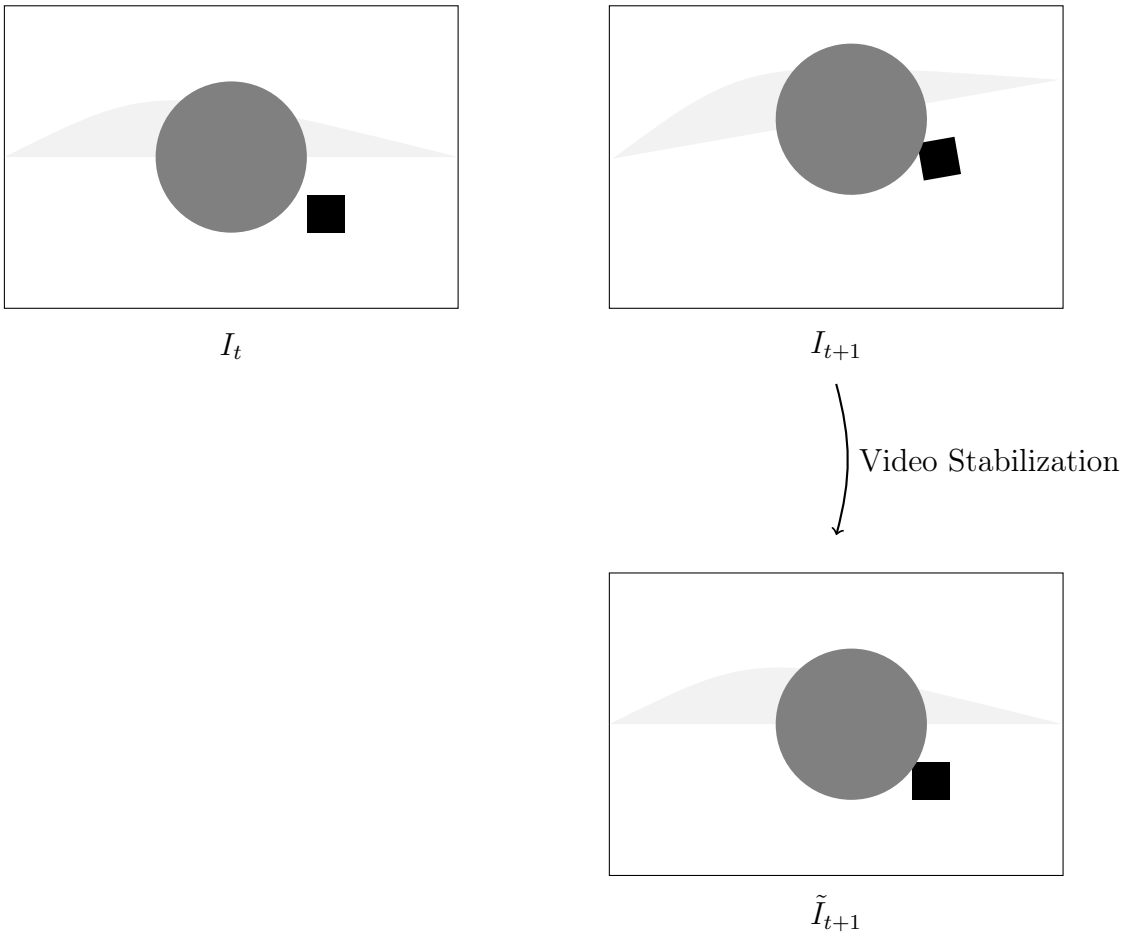

Figure 1.3 Principle of video stabilization. From frame I_t to I_{t+1}, the camera moves, as well as the gray circle in the foreground. Video stabilization consists in computing I_{t+1}, a new version of frame I_{t+1} in which the static objects (here, the background) are motionless.

1.2 Basic notions on video analysis and processing

This section describes basic notions related to video processing, that will be useful in the following manuscript.

1.2.1 Digital representations of videos

First, let us consider how videos are represented in digital format. Videos are sequences of images, called frames. Each of these frames show what the observed scene looked like at a given time t. For the sake of simplicity, we will refer from

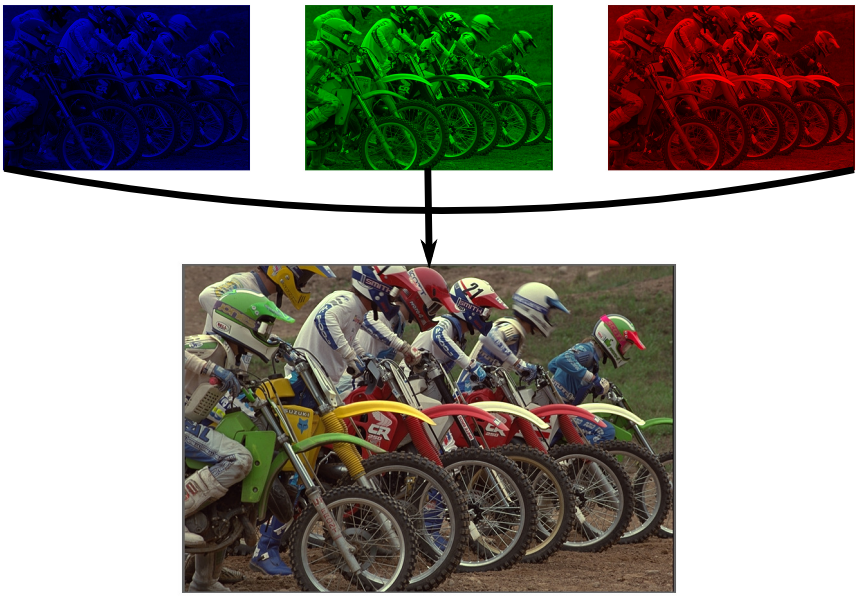

Figure 1.4 Illustration of the composition of a colour video, with the three different colour channels.

now on frame t the frame corresponding to the view of the scene at this time t. The succession of frames gives the impression of continuous movements from the series of still images. This is because the rate of succession of the frames is very rapid. This rate is noted in frame per second (fps), with most videos using 25-30 fps. In digital format, frames are represented as $H \times W \times C$ matrices. H indicates the number of rows and W the number of columns in the matrix. Meanwhile, C represents the number of channels used. In the case of black and white videos, only one channel, representing the luminosity, is used. In colour videos, three channels are used to code the image in the colour-space RGB for the raw format. Figure 1.4 shows the composition of an RGB image. Cases with $C>3$ are possible, such as videos taken with depth cameras, but are beyond the scope of this thesis. Each intersection of a row and column is called a pixel, for picture element. The dimensions $H \times W$ of the frames composing a video is called the resolution, and indicates how precisely the scene can be rendered.

1.2.2 Pinhole camera model

The acquisition of frames is done by projecting the 3D points of the scene onto the 2D camera plane. The link between the 3D scene coordinates (x^s, y^s, z^s) and the camera plane coordinates (x^v, y^v) are described using the pinhole camera model

(see Figure 1.5). If (C, X, Y, Z) is the coordinate system of the 3-dimensional space and (c, u, v) the coordinates of the camera plane onto which the image is projected, the relationship between three dimensional points and their two dimensional projection can be written as :

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = K \begin{bmatrix} x^s \\ y^s \\ z^s \\ 1 \end{bmatrix} \tag{1.1}$$

with the relationship between the coordinates of the camera plane and the vector (U, V, S) given by:

$$\begin{aligned} x^v &= \frac{u}{S} \\ y^v &= \frac{V}{S} \end{aligned} \tag{1.2}$$

In this equation, K is a 3×4 matrix that describes the mathematical relationship between the coordinates of a point in three-dimensional space and its projection onto the image plane. In the simplest case, K only depends on the focal distance f, that is the distance between the image plane and the camera center C:

$$K = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.3}$$

However, this assumes that the aspect ration of the pixels is 1:1. Furthermore, it is only valid if the center of the image plane is also the origin of the coordinate system, when in practice we often use the bottom left corner of the frame as the origin. In such cases, the matrix K is described as:

$$K = \begin{bmatrix} -fk_x & 0 & x_0 & 0 \\ 0 & -fk_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.4}$$

In such cases, k_x and k_y indicate the aspect ratio of the pixels, while x_0 an y_0 indicate the coordinates of the new origin in the current coordinate system. The coordinates of the scene use the camera aperture as the origin, hence any motion of the camera has repercussions on the coordinates of the scene and therefore on the projected image.

1.2.3 Motion blur and rolling shutter

This acquisition is done over a slight time lapse, which can lead to sufficiently fast objects to change projection between the beginning and the end of this lapse,

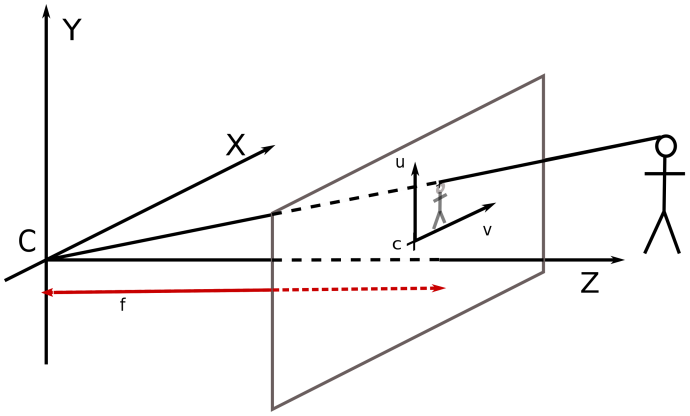

Figure 1.5 Illustration of the projection onto the image plane. (C, X, Y, Z) is the 3-D coordinate system of the scene, (c, u, v) the 2-D coordinate system of the image plane and f the focal distance.

Figure 1.6 Illustration of the rolling shutter effect. Note the tilted tower in the second frame. This is caused by fast camera movements from left to right, which causes vertical edges to be seen as diagonal as the camera changes position between capturing successive rows.

creating what is called “motion blur". While CDD sensors capture the whole frame at the same time, CMOS sensors use instead what is called “rolling shutter": that is, frames are acquired one row at a time rather than all at once. Because of this, fast camera movements can cause deformations in vertical structures, as the camera changes position while the frame is captures. Figure 1.6 shows an exemple of rolling shutter artifact.

1.2.4 Compression and encoding

Once video frames have been captured, they are encoded in specific formats, usually with a degree of compression. The videos considered here are either in AVI or MP4 format with a variety of codec, the h264 codec being the most common. The h264 codec codes the images in the Y’CbCr color space, which uses the

luminance Y' and the luminance with the blue and red channels subtracted (Cb and Cr respectively). However, for the treatment of video stabilization methods, frames are decompressed and converted to RGB format before being treated. The impact of compression on the stabilization results are beyond the scope of this thesis.

1.3 Contributions and publications

The contributions of this thesis are composed of three main parts :

- **A didactic and structured overview of video stabilization methods and current challenges.** The main purpose of this contribution is to provide a fairly unifying framework to allow a better understanding of the progress of this research subject with appreciable industrial and academic benefits. This overview is focused on the main challenges, practical aspects and mathematical core concepts of the video stabilization techniques. By using a step-by-step approach, the video stabilization pipeline can be put in perspective so as to compare the main available approaches and discuss the milestones of this research area.

  W. Guilluy, L. Oudre and A. Beghdadi. Video stabilization: overview, challenges and perspectives. *submitted to IEEE Transactions on Circuits and Systems for Video Technology*. 2018.

- **A new method for outlier removal and camera motion estimation.** The estimation of the 2D or 3D camera parameters from feature trajectories is a tricky process since not all movements present in the video give information on the camera motion. While static objects are only affected by camera-induced movements, other objects undergo displacements that are caused by both the camera motion and the movements of the object in the scene. These moving objects need to be separated from the others and removed in order to compute the correct camera path. We propose a novel approach to assess and select the best feature trajectories to use in the camera motion estimation for video stabilization. Unlike standard approaches used for the selection of feature trajectories, we analyze the movement of the feature trajectories through all frames and compute a global weight by considering multiple criteria such as movement and duration.

  W. Guilluy, L. Oudre and A. Beghdadi. Feature trajectories selection for

video stabilization. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*. Rome, Italy. 2018.

- **A new framework for the video stabilization quality assessment.** To the best of our knowledge there has been very few studies dedicated to performance evaluation of video stabilization methods. The lack of such studies is mainly due to the fact that video instability, like other spatio-temporal distortions and artifacts, is very difficult to model. Indeed, the way this distortion affects the perceived quality is misunderstood and there is no way on how to quantify, in an effective way, the effect of this distortion on the overall quality of the video. Our contribution in this context is twofold. First, by scrupulously reviewing all existing metrics and describing the assumptions behind them, we provide one of the first study dedicated to evaluation. Secondly, we confront existing metrics with subjective results collected on viewers so as to enlighten the existing links between objective scores and visual inspection.

  W. Guilluy, A. Beghdadi and L. Oudre. A performance evaluation framework for video stabilization methods. In *Proceedings of the European Workshop on Visual Information Processing (EUVIP)*. Tampere, Finland. 2018.

1.4 Overview of the manuscript

The manuscript is organized as follows: chapter 2 presents an overview of previous methods of video stabilization. Chapter 3 reviews the different ways that video stabilization methods have been evaluated and presents an investigation of the performances of several video stabilization methods on a sample of those metrics, comparing them to a user study of the stabilization methods to investigate the links between the proposed metrics and the user experiences. Chapter 4 present a novel selection method to identify motion caused solely by the camera motion and evaluates the impact on a standard stabilization pipeline. Finally, chapter 5 offers concluding remarks on the contributions of this work and the current state of video stabilization research.

Chapter 2
Video stabilization : challenges and methods

Chapter 2

Video stabilization : challenges and methods

Video stabilization 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 2.1. 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 is finally processed by using the softened camera movements, so as to generate the stabilized video signal $\tilde{I}$.

As mentioned in Chapter 1, video stabilization has been a major field of research during the last two decades due to the wide range of its potential applications. 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 and to improve the performances. In this article, we provide a structured overview of the stabilization methods proposed in the literature, according to the chart-flow presented on Figure 2.1. 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. These functional blocks have been picked according to their popularity in the community and to their abilities to enlighten the different directions

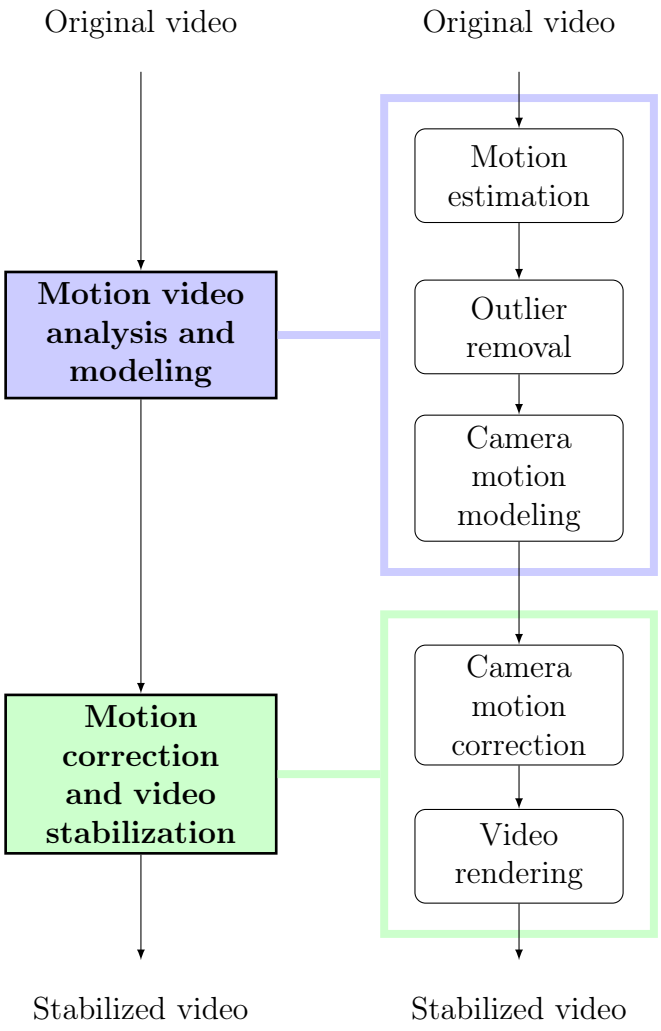

Figure 2.1 Main steps for video stabilization

taken in the literature. 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 the analysis of the video. The aim is to estimate the camera motion from all the motions observed in video. In the following, we decompose this analysis stage into three main successive steps : motion estimation, outlier 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**. Based on general assumptions on the possible impacts of camera movements on the video, this block discards all movements that are not consistent to a plausible camera displacement. Consequently, this step extracts the movements that will be useful in the stabilization process. Finally, the last block provides a **camera motion modeling** from the relevant movements extracted from the video. As will be seen in the next section, this can be done either by assuming a geometrical model for the camera displacement (in this case the block outputs geometrical parameters for 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 by the first stage go through a correction process so as to reconstruct a more pleasing video. In the following, we decompose this processing stage into two main successive steps : camera motion correction and video rendering. The first block performs a **camera motion correction** by smoothing and filtering 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 rendering** step. This final step reconstructs a new video with smoothed camera movements.

In the following, for the sake of completeness and clarity, the video stabilization process is presented as a set of interdependent steps described in the order they appear in the whole VS pipeline. First, the video analysis step consists of estimating motions present in the video (Section 2.1), then identifying those resulting from the motion of the camera (Section 2.2) to be used to model the original path of the camera (Section 2.3). The video is then stabilized by correcting the path of the camera (Section 2.4) and rendering a video (Section 2.5), simulating the scene as captured by a virtual stabilized camera following the corrected path.

Notations

The aim of this 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 $\mathbf{z}_t$.

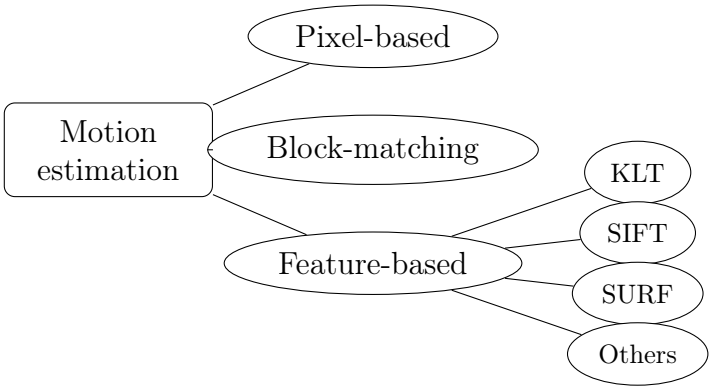

Figure 2.2 Main approaches for motion estimation

2.1 Motion estimation

The first step of any video stabilizer is to acquire knowledge of the camera movements. However, in the context of digital stabilization, no direct information about the camera is known, since only the video sequence is available. Motion estimation aims to recover movements present in the video sequence, which will be used to determine the motion of the camera. These movements are estimated by searching for correspondences 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 if they are assumed to correspond to the same element within the captured scene. Several approaches have been introduced to solve this problem: some try to find a match for every pixel in the frame (Section 2.1.1), others use blocks of pixels (Section 2.1.2) and finally, points of interest can be used to estimate the movements (Section 2.1.3).

2.1.1 Pixel-based matching

Pixel-matching methods seek to determine the motion of pixels between two frames [19]. As illustrated in Figure 2.3, 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 by the pixel of the following frame corresponding to the projection of the same 3D point onto the new camera plane [20]. To determine this point-to-point correspondence, the luminance of any given object is assumed to be constant throughout a video sequence. Therefore, such methods attempt to match pixels with the same intensity. However, many pixels in a given pair of frames may have similar intensity; therefore additional constraints are needed to obtain a unique solution [21], [22]. Early methods suppose that the movements

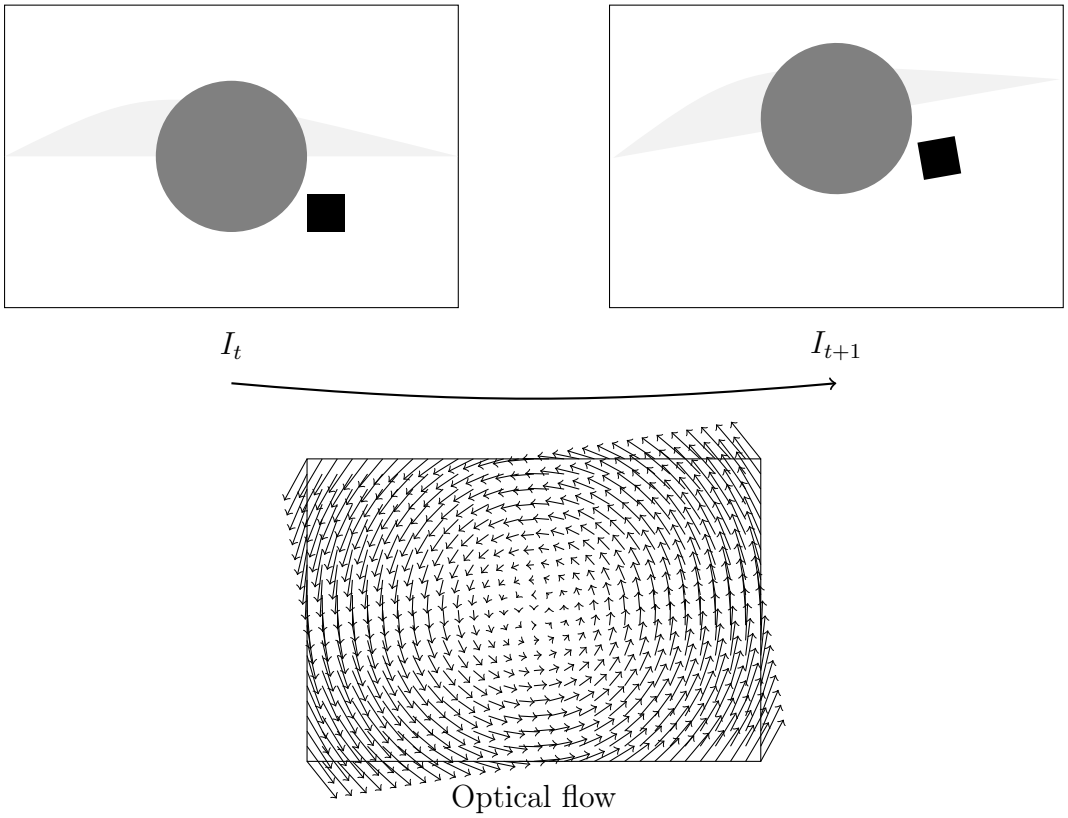

Figure 2.3 Principles of optical flow. Optical flow aims at computing the displacement of each pixel between frame I_t and I_{t+1}. It is often displayed as a vector field.

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 parameters that best explain the overall displacement of the frame. Those parameters are easily estimated by minimizing the luminance difference between the two adjacent frames [23].If H_t denotes the transformation matrix between the frames t and $t+1$, the best 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 [24]. This approach solves simultaneously the different steps of the camera motion modeling. This saves times, but needs a pre-determined motion model. Since only adjacent frames can be compared, there is not enough information to solve a 3D motion model, constraining the results to 2D models. In addition, this cannot be used with an outlier removal scheme. It is also sensitive to changes in illumination. A recent method [25] uses a similar approach to identify the correction directly, using the pixel matches as one component of an energy function using neural networks.

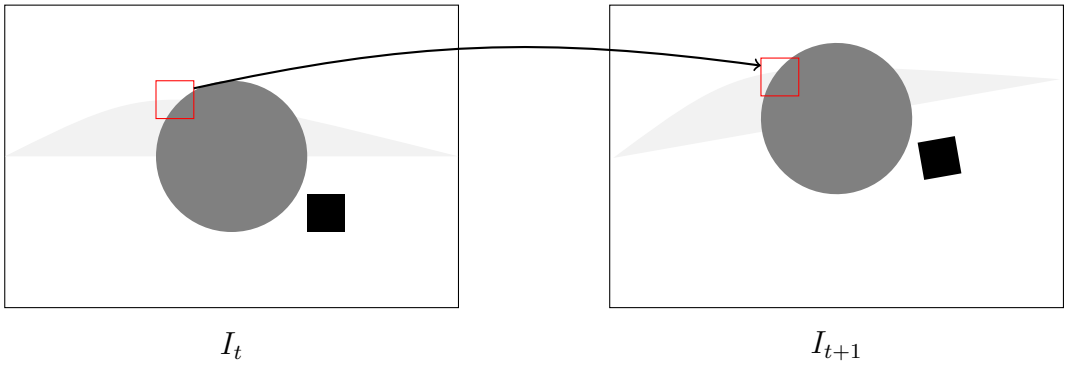

Figure 2.4 Principles of block matching. Block matching aims at computing the displacement of blocks of pixels between frame I_t and I_{t+1}.

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 that neighbouring pixels have similar movements [26] or that the flow is smooth or piece-while smooth [27]. The flow is often computed iteratively using the spatial and temporal gradients[28]. The main advantage of using optical flow is that it recovers a dense flow field, which is necessary for some stabilization methods [29] or for enhancements such as in-painting [30]. Another advantage is that neighbourhood relations are easy to determine between motion vectors, which can be exploited in later steps [31]. However, computing dense optical flow may require heavy computations. This can be partially alleviated by using other faster matching methods to compute an initial flow before running the iterative algorithm [32]. 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. The main drawback of optical flow is the running time: this is often the longest operation in a video stabilization process that uses it, as the movements from easily matched features must be propagated to flat spaces. Liu et al. [29] report 1.1 seconds per frame to compute the optical flow out of 1.5 seconds per frame for the whole stabilization algorithm. In addition, times is spent calculating the optical flow on features that have their own movement in addition to the camera's, and therefore are hard to use to evaluate the camera's movements. Finally, because motion vectors are determined for pixel coordinates and not for specific features, trajectories of objects or features cannot be determined.

2.1.2 Block-matching

Instead of finding correspondences between pixels, block-matching methods use blocks of pixels of size $(2n+1)\times(2n+1)$ and estimate their displacements between adjacent frames, as illustrated on Figure 2.4. The use of blocks allows to remove some ambiguities that appear 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 only consider blocks of pixels within a certain search radius of the block to be matched, further restricting the possibilities for matches. The best displacement $(u_x, u_y, 1)$ for a given pixel $(z_t = (x_t, y_t, 1))$ is the one minimizing the energy E , with the considered displacements $(u_x, u_y, 1)$ smaller than the given search radius.

$$E(x_t, y_t, u_x, u_y) = \sum_{k=-n}^{n} \sum_{l=-n}^{n} ||I_t(x_t + k, y_t + l, 1) - I_{t+1}(x_t + u_x + k, y_t + u_y + l, 1)||^2 \tag{2.1}$$

To that end, block-matching approaches are based on search windows that constrain the possible motions. This is computationally very efficient, but two main problems emerge. First, the aperture problem caused by the search windows: smaller windows run the risk of being too small to contain the true motion, while larger windows are prone to containing several possible matches. Secondly, homogeneous surfaces have little to promote one match over another. These regions are therefore often dropped from computing the displacement, and are considered ambiguous. The result is a motion field presenting similar advantages to the optical flow fields, with easy neighbourhood relations but no object trajectories. The main disadvantages is that untextured areas are not filled in.

2.1.3 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, as illustrated on Figure 2.5. By processing the entire video frame after frame, the positions of these points can be tracked using the properties of the given features, forming trajectories. One of the advantages of using this approach is that the same point can be tracked and recognized across many frames, from the frame it is originally identified to the frame where it is no longer present. In particular, the study of the trajectories can give a better insight on the movements present in the scene.

A commonly used [10], [11], [13], [14], [17], [18], [33], [34] tracking algorithm is the KLT tracker [35]. A feature detection algorithm is used to initialize the position

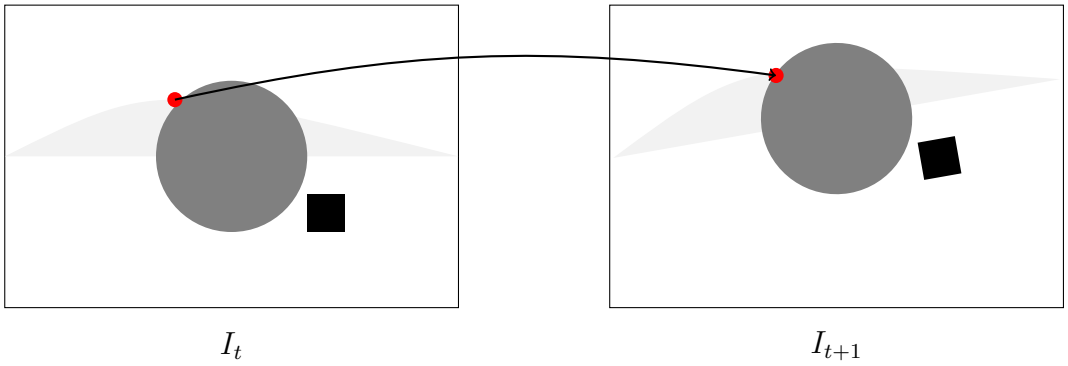

Figure 2.5 Principles of feature point matching. Feature point matching aims at computing the displacement of only a subset of relevant pixels between frame I_t and I_{t+1}.

of tracked points, which are then tracked using optical flow. A confirmation check verifies that the feature has been tracked correctly, otherwise the feature trajectory is ended. New feature points can be detected in later frames if too many feature points have been lost.

SIFT points are also widely used [2], [36]–[39]. These features use descriptors based on the image gradient to obtain very specific descriptors that make matches very reliable. The descriptors 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 [40], but optimized for speed, making them a good alternative [41]–[43]. SIFT features have also been used in conjunction with line detection methods [44], as deformations caused by stabilization are particularly visible on lines.

Other features used include Maximally Stable Extremal regions (MSER) [45] or FAST corners using BRIEF descriptors [12], [46]. MSER detect contiguous regions whose borders are darker/brighter than any pixel in the region. An ellipse is then fitted over the region, using the covariance matrix to identify the ellipse axis in order to make it rotation-invariant. FAST corners are detected when a contiguous number of pixels are darker or lighter by a tolerance threshold compared to the central pixel, and descriptors are based on binary comparisons between the central pixel and the rest of the patch. FREAK descriptors [47] have also been used in some recent works **Zhao2019TrajDerivatives. These binary descriptors use overlapping sa** [48], [49]

Feature-matching provides accurate and fast results, and the obtained trajectories allow for additional temporal analysis in the remaining steps of the process,

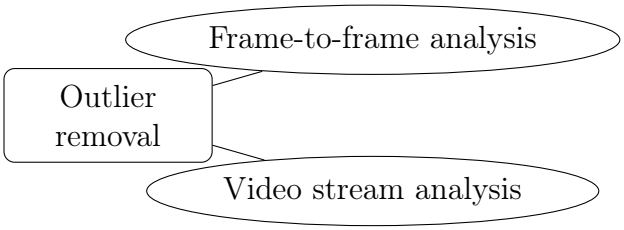

Figure 2.6 Main steps for outlier removal

although scenes with large uniform regions can sometimes yield few features per frame. However, because features are spread unevenly across the video frames. This can lead to some areas being over-represented in the motion analysis. Furthermore, neighbourhood relations are harder to establish.

2.2 Outlier removal

While all movements observed in the video sequence are affected by the motion of the camera, not all observed movements are suitable to determine the camera motion. The presence of moving objects in the filmed scene can be a source of errors, as the movements of the objects could be mistaken for those caused by the camera motion. 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 after the movement estimation [50], that are designed to remove these outliers. Two main approaches can be used, that are either based on frame-to-frame analysis (Section 2.2.1) or on the whole video stream (Section 2.2.2).

2.2.1 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 [51]. Using a given motion model, 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 selected. Different variants on RANSAC have been used. For instance, umLESAC uses preliminary tests before measuring the fitting error to discard bad data samples quickly and adapts the number of iterations to the dataset [52]. Another variant, ORSA, uses an a contrario approach to avoid a hard threshold on the fitting error [53]. This approach has the advantage of solving for camera motion and detecting outliers at the same time. But it can led astray if the majority assumption is false for a single frame.

Assumptions on camera motion can also be used to determine outliers. One hypothesis is that objects in the scene move faster than the camera. Outliers are detected simply by thresholding the velocity of the observed movements [45]. Another hypothesis, in the case of dense optical flow, is to consider the smoothness of the flow field. Thresholding the spatial gradient of the vertical and horizontal flow fields detects the edges of moving objects, and by successive iteration can remove all flow vectors corresponding to moving objects [29]. Median filters can also be applied, on both fine scale to remove tracking errors and small moving object, then on a larger scale to remove larger outliers [54]. Finally the spatial distribution can be taken into account. For instance, a RANSAC variant applies a grid over the reference frame and limits the number of movements selected from one quad for any iteration of RANSAC [34], which avoids over-fitting the model to a specific area of the frame. Using a similar grid, RANSAC can be applied separately to each quad, resulting in local outlier detection [4]. Because only adjacent frames are used, this approach can be applied in real-time without requiring a buffer. It can also be used with any type of motion detection methods.

2.2.2 Video stream analysis

Outlier detection can also take into account motion over more than two frames. This allows the consideration of 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. One criteria that can be exploited is the difference between expected and observed motion. In particular, the motion induced by the camera has been modeled as a projection into a low rank subspace. Trajectories whose projections differ strongly from the original motion at any given time are considered faulty and discarded [18]. similarly, if the RANSAC algorithm has been used to determine an initial transformation, then the differences between the known positions of features and the expected positions can be computed [48]. These differences, called projection errors, can be used to refine the outlier rejection. Trajectories that repeatedly cause large

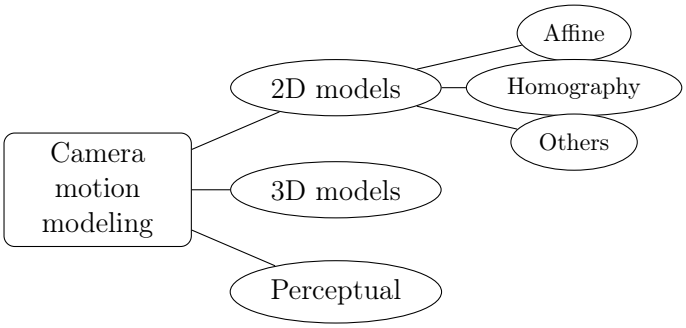

Figure 2.7 Main approaches for camera motion modeling

projection errors can then be reclassified as outliers. In addition, trajectories that have been often classified as outliers can be excluded from the initial RANSAC execution and reclassified as inliers only if they have small projection errors. Observing trajectories over a given temporal window can also give insight on which trajectories are the most reliable. Firstly, some approaches require trajectories of a minimum length, such as models using epipolar geometry [55], which require temporally distant frames to work. Trajectories that do not span the required frames are therefore considered outliers. Methods that work directly on trajectories may require a minimum length for a trajectory to be usable, and need to augment some trajectories that are too short, in which case it is logical to prioritize the trajectories requiring the least degree of interpolation [11]. Finally, moving objects often leave the frame quickly as they pass through the scene, so longer trajectories are prioritized as more likely to belong to the static background of the scene [18]. The spatial distribution can also be exploited over a period of time. Bi-layer clustering is a method used to detect large moving objects in the foreground of videos. It uses motion and colour to segment feature trajectories into two clusters, and chooses to remove the cluster with the greater compacity, as the background of a scene is far less compact than moving objects [11]. These methods are only applicable if feature trajectories have been obtained using feature-matching while detecting motion in the video.

2.3 Camera motion modeling

Once 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 2.3.1). By recovering the depth information and using 3D models, it is also possible to seek for the original 3D displacements of the camera (Section 2.3.2). Alternatively, another approach is to avoid geometrical models to obtain perceptually plausible models, in order to obtain visually acceptable corrections rather than physically accurate ones (Section 2.3.3).

2.3.1 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 $\mathbf{z}_t \triangleq [x_t, y_t, 1]$ belonging to frame t, its coordinates $\mathbf{z}_{t+1}$ in frame $t+1$ are given by

$$\mathbf{z}_{t+1} = H_t \mathbf{z}_t \tag{2.2}$$

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

$$H_t = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}, \tag{2.3}$$

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 [16]. 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 [24], [56]. Moreover, the 2D assumption is often valid on a local temporal scale when the movements of the camera are not to large. They can also be computed from frame-to-frame correspondences, and can thus be used in conjunction with any type of motion detection. Finally, the parameters of the transformation and the pixel position are enough to determine the expected location of the pixels after applying camera motion. This means that the motion caused by the camera is known for every pixel and the corrections required can likewise be known exhaustively.

The simplest parametric planar transforms to be considered are the similarities (also referred to as simplified affine models) [16]. They are able to handle translations, scaling and rotation along the camera axis and are based on only four parameters a, b, t_x, t_y such that

$$H_t = \begin{pmatrix} a & -b & t_x \\ b & a & t_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.4}$$

Parameters a and b handle the scaling the rotation along the roll-axis, while t_x and t_y respectively model the horizontal and vertical translations. By setting $a = \lambda\cos(\theta)$ and $b = \lambda\sin\theta$, the scaling/rotation effects can be specified with λ the scaling parameter and θ the rotation parameter [57]. The estimation of the four parameters can be solved by linear Least Squares Method on a set of redundant equations [36], [58], [59], possibly combined with filtering/outlier removal [32], [42]. Histogram approaches have also proven to be effective in this context [60], [61]. 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 place perpendicular to the z-axis [56]. These results allow to think that by only considering scale, z-axis rotation, and translations, it is possible to obtain an acceptable approximation [6], since the impact of pitch and yaw rotations on the final image warping are often minimal 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 little deformations, which makes it robust to outliers and noise [62]. However, it cannot account for strong rotations outside of the camera axis , which may limit its performances in strongly degraded videos.

Slightly more complex with six parameters, the affinity (or generalized affine model) is the most commonly used 2D model. By replacing parameters a and b by four parameters $a_{11}, a_{12}, a_{21}, a_{22}$ the matrix transformation writes

$$H_t = \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.5}$$

The affinity model encompasses most of the qualities of the similarity model, but additionally allows the possibility of shear [9], [37], [44], [46]. The parameters can be estimated with standard differential motion techniques [23], with more complex cost functions [24] or through multi-scale [63] or hierarchical analysis [30], [64]. The major advantage of using an affinity model lies in the fact that it naturally handle global motions, for which the affinity parameters at every

location should be the same [65]. The model can deal with scenes containing small relative depth variations and zooming effects [38] and provides an acceptable compromise between accuracy and computational cost[45]. However, being a 2D planar transform, it cannot model non-linear inter-frame motion [4].

Finally, the most exhaustive 2D is the homography, which uses 8 parameters. The matrix transform becomes

$$H_t = \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ a_{31} & a_{32} & 1 \end{pmatrix}. \tag{2.6}$$

While the interpretation of the coefficients is not as straightforward as for similarity or affinity [2], they control rotations, translations, zooming and sheering in the x- and y- axis [66]. Homographies have been popular for image registration [53], and most authors use techniques developed in this context for the estimation of the eight parameters [13], [14], [44]. Nevertheless, the homography model has the potential to cause severe deformations, particularly in the presence of outliers [12].

Other 2D models include simpler models with 3 [62], [67], [68] or 4 parameters (2 rotations and 2 translations) [69]. So-called 2.5D models propose to compromise between 2D and 3D models, by considering cases where 3D displacements can be simplified to avoid the need for depth (translations 1 axis (x, y or z)) [70].

2.3.2 3D models

Contrary 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 are able to provide physically realistic displacements in all the directions. Their ability to compute a precise estimation of the movement is dependant on the depth recovery step, i.e the estimation of the distance to the camera of each 3D point seen in the frames [55], [71]. Recovering depth consists in analyzing the original video (where the available information lies in 2D planes), in order to retrieve the original 3D content of the scene. This task, referred to as structure-from-motion [17], 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, that insure that sufficient motion is present. This is only possible using feature trajectories, as it is the only

motion estimation method that can match points across distant frames. Even so, 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 [72], manifold constraints [73]) that help to provide accurate computation. The computation cost, which is often high, can be be kept under control by only focusing on particular regions of interest [74], [75]. Because of this, the depth is usually only recovered for certain pixels, which leads to incomplete motion fields and corrections.

The main drawbacks of 3D models can also be taken into account 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 only consider certain displacements in the 3D space. For instance, by considering only rotations, it is possible to drop the depth recovery task and only focus on the estimation of the calibration matrix and the rotation matrix [7], [10]. This assumption appears to be valid for hand-held shakes but is violated in more complex conditions 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. By using these constraints, it is possible to simplify the general 3D model and provide ad hoc formulation that are lighter than structure-from-motion [8]. Finally, some authors propose to introduce the notion of 2.5D models, and to define ad hoc models that correspond to classical situations (dolly, vertical or horizontal tracking...). The specification of the movements helps to compute the depth estimation and can therefore provide accurate results.

2.3.3 Perceptual models

As seen in the previous subsections, there are many different motion models to choose from. The choice of the appropriate model can be tricky when no extra information is available on the camera movement, which unfortunately is often the case. Moreover, 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 [18]. Several approaches have been proposed to try to combine the robustness and computational efficiency of 2D models with the accuracy of 3D models. Such approaches have in common an important principle : the main objective of video stabilization is to improve visual comfort. As a result, these methods prefer to avoid geometrical models and instead use models that provide visually plausible videos rather than physically accurate ones. Such models are referred to as perceptual models. While their

principles vary, they have in common that they strive for results that are not necessarily geometrically accurate but will seem accurate to human perception.

One approach is to apply several 2D transforms to a single frame.. 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 [76]. An initial global homography is used to fit one frame onto the next. Then, each frame is divided into 4x4 quadrants, and each quadrant of the reference frame is fitted to the corresponding quadrant of the following frame using a different homography. These localized homographies can account for different motions (such as those caused by parallax) in a single frame. This approach allows for more flexibility while retaining the robustness of 2D transforms, but no longer corresponds 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 [77]. By projecting the trajectories recovered from feature points into a low-rank subspace through matrix factorization techniques, it is possible to model the motion of the camera without requiring knowledge of the depth, while retaining a flexibility that would be lost with a fixed 2D motion model [18]. The projected trajectories are known as eigen-trajectories and can then smoothed like any other trajectories. The projection is done by computing the concatenating the feature trajectories and applying the SVD decomposition to the resulting matrix. Truncating the SVD decomposition to the k largest singular values corresponds to the projection in a k rank subspace. Since projecting the trajectories in their entirety is impractical, a moving factorization strategy is used. An initial projection is done over a given number of frames, and the rest of the trajectories are progressively factorized into the initial subspace. Tang et al. use the same principle but use alternative factorization methods either based on a sparse representation strategy to improve the factorization [78], or using local projection rather than projecting all trajectories in the same subspace [79]. Once the factorization is complete, we obtain three sets of parameters : the singular values, parameters that depend on the feature tacked and parameters that depend on the frame considered. These last parameters can be treated as the camera parameters for the purposes of smoothing the trajectories. This approach requires feature trajectories, and is liable to fail if few long trajectories are available.

It is also possible to forego any motion model and treat the observed trajectories directly. Several methods apply filtering or path-fitting to the feature trajectories and then rely on sparse reconstruction methods [11], [33]. Koh et al. [11] find

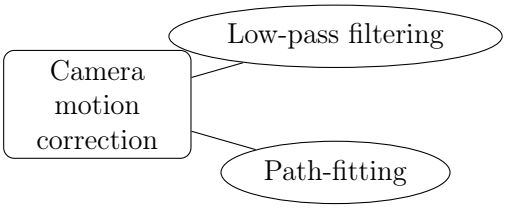

Figure 2.8 Main approaches for camera motion correction

an optimal trajectory for each feature individually, and find a reconstruction step that will find a good compromise between the obtained optimal trajectories. Wang et al. [33] use neighbourhood constraints to determine the optimal paths for the different features. Similarly, this can be done for all pixels using optical flow in successive frames to obtain the trajectories of all pixels in a given frame over a time window, and smoothing the resulting motions [31]. Another approach is to consider the motion at a given location rather than tracking pixels or features. Using optical flow, it is possible to observe the variations of motion at any given frame coordinates over time. The motions occurring at given locations are termed "pixel profiles", and the same stabilization criterion can be applied to obtain a stabilized video [29], [80].

2.4 Camera motion correction

Once the camera motion has been modeled, new camera movements should be determined that will result in a better video. This begs the question : what types of camera motion should be used instead of the originals? Since one of the most problematic aspects of camera motion is the high-frequency shakes, which causes considerable visual discomfort, filters are often used to remove such problems (Section 2.4.1). Another possibility is to look to cinematographic considerations for the type of motion used (Section 2.4.2).

2.4.1 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 obtained the stabilized position of feature points, or to the camera motion parameters to obtain the stabilized position of the camera. In the latter case, parameters are generally treated separately. A very common filter is the Gaussian filter [10], [30], [31], [38], [45], [64]–[66], [72] (see Figure 2.9). It suppresses the high-frequency motion that are the most detrimental to visual comfort, and

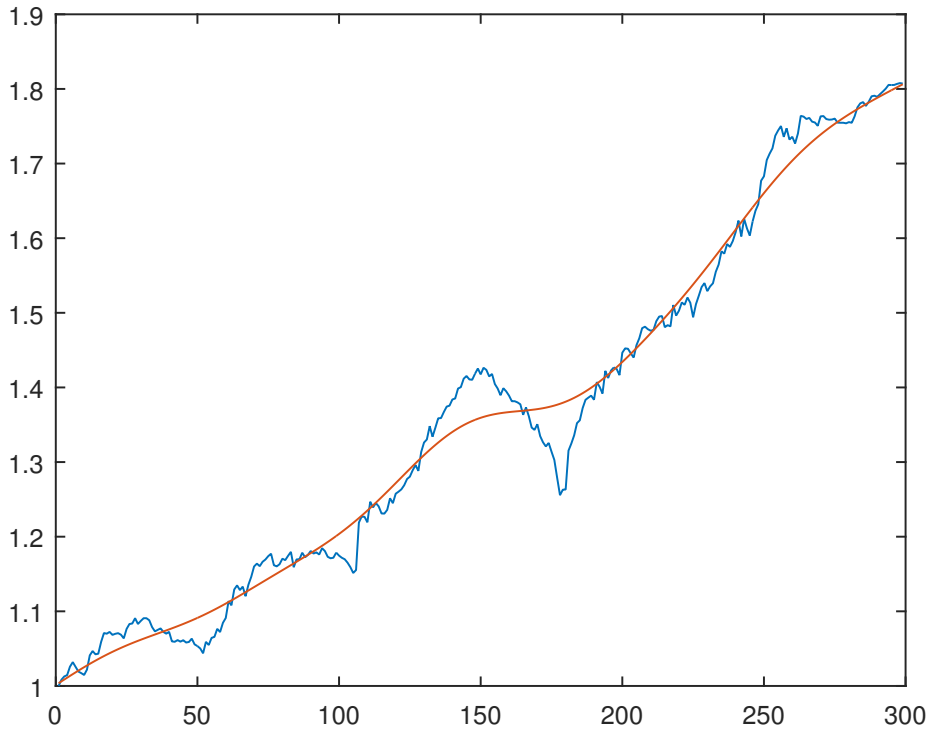

Figure 2.9 An example of camera motion filtering on the first camera parameter with a gaussian filter. The original parameters are shown in blue, whereas the filtered parameters are shown in red.

the level of stabilization is easy to modify by altering the width of the Gaussian kernel used. Another common filter is the Kalman filter [8], [14], [24], [32], [46]. It uses the observed motion to estimate the intentional motion and the unwanted motion to be corrected. Motion Vector Integration [41], [57] 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 [43], [70] with cut-off frequencies based on the considered applications.

How these filters are applied depend on the way the camera motion is modeled. In most instances, the filters are applied separately to each parameters of the camera motion model. Otherwise, trajectories are filtered.

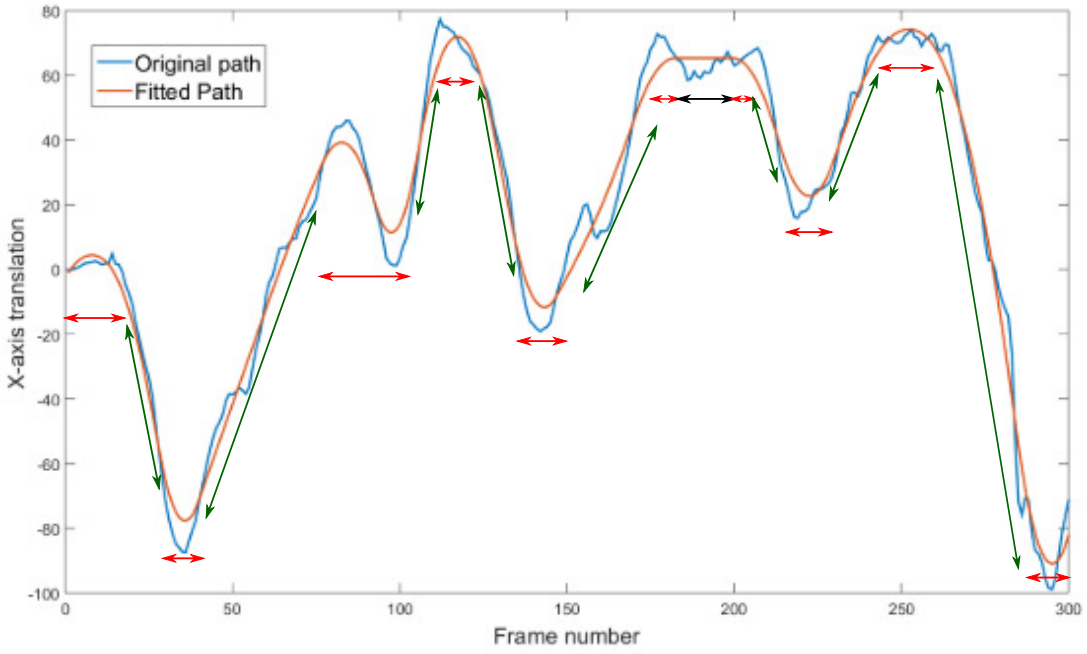

Figure 2.10 An example of path-fitting. The x coordinates of a trajectory over 300 frames is depicted in blue. The coordinates of a stabilized version of this trajectory is depicted in orange. The black arrow indicates a static segment identified by the path-fitting algorithm. Green arrows indicate where steady movements have been detected, and red arrows show the frames where constant acceleration was enforced.

2.4.2 Path-fitting

Determining what is a "good" path for the camera motion is difficult. Several criteria can be taken into consideration, such as quality of motion and the introduction of artifacts. Quality of motion can be simply considered as the smoothness of the movements, but we can also look to cinematographic criteria to have an idea of which type of movements are considered desirable. In general, we try to obtain one of three types of camera movements: still shots with no movements, tracking shots with constant movements and smooth transitions between different segments. The transitions are considered smooth when the acceleration is constant. These types of motion are illustrated in figure **??**. The main artifact caused by video stabilization is a loss of resolution. Stabilizing a video entails simulating a camera path different from the original. Because of this, parts of the scene as filmed by the simulated camera may not have been filmed by the original camera, leading to undefined areas. The most common way to deal with these areas is to crop them out, leading to a loss of resolution.

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 choose which motion type is expected [71]. However, these models do not hold for long video sequences, so the quadratic models can be replaced by spline interpolation, with control point chosen by the user [18]. 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 [38]. 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 juncture between static and tracking segments. Another approach is to use an energy minimization scheme to determine the new camera motion[11], [33], [76]. The regularity of the camera motion is represented by one or more energy terms depending on the expectations on the camera motion. The loss of resolution is either used as a hard constraint on the smoothed camera path or as another energy term. Other considerations are easy to implement as additional energy terms or constraints. Grundmann & al [4] proposed three energy terms for the motion regularity using L1 optimization, based on the first, second and third order derivation of the camera motion. The first order derivation corresponds to the expectation of a static camera, the second order derivation to the expectation of constant camera motion and the third order derivation to the expectation of smooth motion variation. Using L1 optimization forces the solution into one of these roles rather than finding a compromise. This minimization is subject to both a maximum deviation from the original path and a minimum coverage of the video frames. Song & al [42] use a similar constraint on the deviation from the original camera path but only use an L2 constraint on the second order derivation of camera motion. Several methods consider the deviation from the original path as an energy term combined with a first order derivation on the camera motion [3], [12], [29], [58], [73], [80]. A second order derivation can also be used for the data term [81], or combine second and first order derivation [11]. Additional constraints can be used to avoid distortion artifacts, with constraints on spatial rigidity [33]. 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 correct [76].

2.5 Video rendering

Once a new camera path has been computed, a new video corresponding to this path must be rendered. This step depends on the choice of camera model. Most

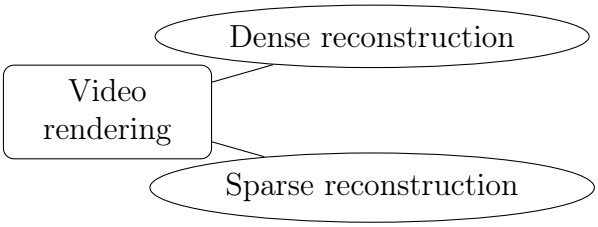

Figure 2.11 Main approaches for video rendering

geometrical models describe the original and corrected motion for each pixel, allowing for dense reconstruction where the all modifications applied are known (Section 2.5.1). 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 2.5.2).

2.5.1 Dense reconstruction

In the case of dense reconstruction, the original and stabilized positions are either known or can be 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 need to find the right transformations to fit the original motion into the corrected motion. Let H_t denote the transformation between frame t and $t+1$ and $\tilde{H}_t$ the smoothed version of this transformation. 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 corrected motion, which can avoid the accumulation of motion estimation errors from distant frames [13]. 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 (SFM) [17]. In the case of perceptual models using dense flow fields, the original optical flow is corrected to the stabilized optical flow.

2.5.2 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) [71] spreads the correction from known pixels to the rest of the frame by applying a 4x4 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 z_t^i is re-written as a function of the enclosing quad vertices:

$$z_t^i = w_i^t V_i[t] \tag{2.7}$$

with $V_i[t]$ is a vector containing the coordinates of the vertices enclosing the i^{th} interest point and w_i^t a vector of weights that sums up to one. This way changing the position of the vertices also alters the position of the enclosed pixels. New vertices positions $\tilde{V}$ are computed using an energy minimization method, with two components. First, the data energy term constrains the known points into their stabilized position.

$$E_{data}(\tilde{V}) = \sum_i ||z_i^t - w_i^t \tilde{V}_i[t]|| \tag{2.8}$$

Second, the structure energy constricts each quad to a similarity transform, weighted by the salience of each quad. This avoids deformations in highly textured areas. To enforce the similarity, quads are divided into two triangles. Under a similarity transform, the location of each vertex can be computed from the positions of the other vertices. Deviations from this position indicate that the applied transform is not a similarity.

$$V_1 = V_2 + R_{90}(V_3 - V2), R_{90} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{2.9}$$

The structure energy of a vertex can then be defined as

$$E_{structure}(\tilde{V}_1) = \tilde{V}_1 - \tilde{V}_2 + R_{90}(\tilde{V}_3 - \tilde{V}2) \tag{2.10}$$

The structure energy of the mesh is obtained by summing up the energy of each vertex. The final energy is the weighted sum of the data and structure energies:

$$E(\tilde{V}) = E_{data}(\tilde{V}) + \alpha E_{structure}(\tilde{V}) \tag{2.11}$$

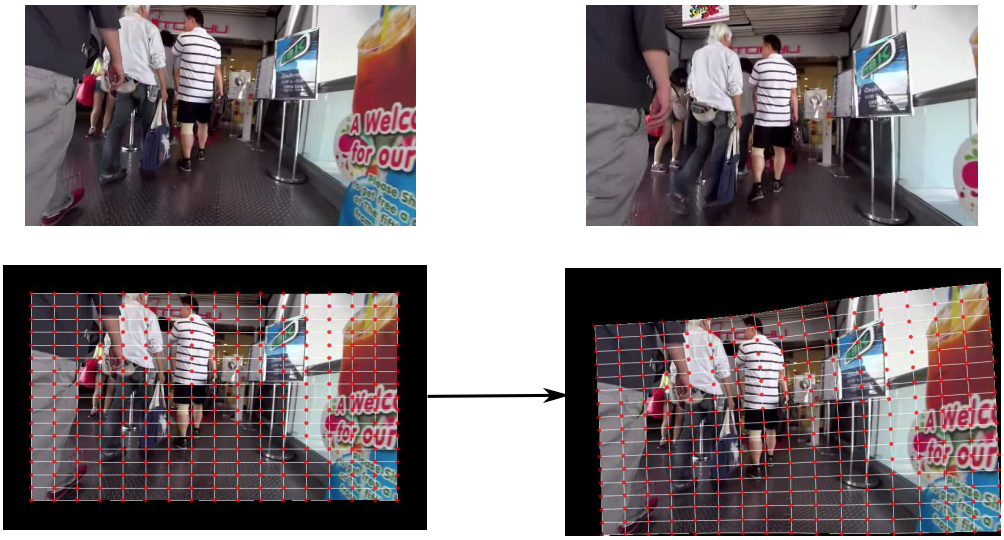

Figure 2.12 An example of sparse reconstruction. The top row show two frames from a video sequence. The bottom row shows the warping scheme from Liu & al. [76], which warps the left hand frame to the right hand one.

The parameter α controls 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 [75], [76] (see Figure 2.12), 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 [33]. The similarity constraint is also changed to an homography constraint. Finally, Koh et al. [11] 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.

2.6 Challenges and perspectives

Video stabilization has seen considerable progress in recent years. However, many challenges remain, in particular regarding the evaluation of the stabilization process. Despite several propositions, there is still no accepted metric to to quantify the quality of video stabilization. Indeed, to our knowledge there is no reference data-set to test different stabilization processes, as different scenes lead to very different difficulties. This is particularly problematic, as video stabilization

necessitates a trade-off between the removal of unwanted motion and the loss of resolution it causes. Without an objective measure, this trade-off needs to be fixed heuristically. 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 determination of the stabilized camera path. Another challenge is the presence of moving objects. Moving objects often result in occlusions, which are challenging to in-painting methods 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. Motion blur, caused by fast camera motion, is also problematic as it disrupts the estimation of motion in the video, which impacts every aspect of the stabilization process. Finally, the choice of motion model is a difficult one, as 3D models are computationally heavy 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. The detection of moving objects, while remaining difficult, has seen promising results by exploiting either motion discontinuities or the evolution of motion over time. While full 3D models remain unstable, 2D models have proven efficient for smaller degrees of stabilization or for scenes lacking parallax, while perceptual models have shown a wider range of stabilization without excessive loss of robustness.

Authors	Motion estimation	Outlier detection	Motion model	Motion correction	Video rendering	+	-
Liu & al.[71]	Features		3D		sparse	handles parallax	SFM not robust
Liu & al.[18]	Features	frame-to-frame	perceptual		sparse	robust to motion parameters, handles parallax	does not handle moving objects
Litvin & al.[24]	Pixel-matching		2D	Filtering	Dense	reduces/eliminates resolution loss	does not handle parallax
Gleicher & al.[2]	Features(SIFT)		2D	path-fitting	dense	fast, good video dynamics	low feature count/moving object cause problems
Grundmann & al.[4]	Optical flow	frame-to-frame	2D	Path-fitting	dense	stable, avoids sudden shifts in case of bad estimations	does not handle parallax, bad trade-off occur
Matsushita & al.[30]	Optical Flow		2D	Filtering	dense	avoids resolution loss and blur	does not handle parallax, bad with moving objects
Liu & al.[76]		frame-to-frame	perceptual	path-fitting	sparse	robust and more precise than regular 2d	motion blur, large moving objects
Chang & al.[3]	optical flow	frame-to-frame	2D	Filtering		fast (real time)	does not handle parallax
Goldstein & al.[34]	Features	frame-to-frame	perceptive	Filtering	CPW	2D/3D hybrid	needs good trajectories and large objects not handled
Liu & al.[29]	Optical Flow	video stream	perceptual	path-fitting	dense flow	handles parallax well	slow, dominant foreground
Yang & al.[62]	Features	video stream	2D	kalman filter		handles abrupt motion	does not handle parallax

Table 2.1 Summary of video stabilization methods (part 1)

Authors	Motion estimation	Outlier detection	Motion model	Motion correction	Video rendering	+	-
Koh & al.[11]	Features	video stream	perceptual	path-fitting	CPW	handles dominant foreground	very specific case for outliers
Sanchez & al.[82]		frame-to-frame	2D			good motion composition strategy	remains in 2d models
Wang & al.[33]		video stream	perceptual	path-fitting	sparse	parallax without SFM	needs long trajectories, bad with foreground objects
Kingsbury & al.[83]	pixel-based		2D	filtering	dense	interpolated frames lower motion blur	no outlier rejection, does not handle parallax
Yang & al.[84]	block-matching	frame-to-frame	2D	static		real-time	does not handle parallax
Farid & al.[23]	use image intensity		2D	static			does not handle parallax
Chen & al.[37]	Features(SIFT)	frame-to-frame	2D	filtering	dense	reduces resolution loss	does not handle parallax
Morimoto & al. [16]	Block-matching		2D	static		fast	very simple model
Tsoligkas & al.[32]	Block-matching	frame-to-frame	2D 2d	filtering			simple model
Zhang & al.[85]	Features	video stream	3D	path-fitting		full motion model	SFM slow and not robust
Rawat & al.[64]	pixel-based		2D	filtering	dense	handles abrupt motion	does not handle parallax
Ringaby & al.[10]	Features	frame-to-frame	3D	filtering		avoids SFM problems	model not always valid

Table 2.2 Summary of video stabilization methods (part 2)

Chapter 3
Performance evaluation of video stabilization algorithms

Chapter 3

Performance evaluation of video stabilization algorithms

3.1 Introduction and motivations

Over the two last decades many methods for video stabilization have been proposed in the literature. However, some open problems have not yet been thoroughly addressed. In particular, the performance evaluation of video stabilization algorithms, or video stabilization quality assessment (VSQA), remains an open problem. Very often the quality of the processed video is evaluated using some subjective and intuitive criteria or by using some simple quantitative measures. However, these quantitative measures do not exploit any knowledge nor well defined model of the visual discomfort due to video instability. To the best of our knowledge there has been very few studies dedicated to performance evaluation of video stabilization methods [86]–[88]. Because of this, while several metrics have been proposed to validate video stabilization algorithms, there are no agreed-upon criteria or metric for assessing the quality of video stabilization.

This evaluation is a complex problem for several reasons. Firstly, the goal of video stabilization is to remove unintentional camera motion, but the definition of the intentional and unintentional parts of the movements observed in the video is inherently subjective. A formal definition would require a priori knowledge on the original intention of the cameraman during the video capture, which is often unknown or difficult to predict. Therefore, one has to rely on some heuristics and intuitive definitions to discriminate between the two types of camera movements, namely "unwanted" and "wanted". However, the impact of spatio-temporal distortions or artifacts, such as those caused by the motion of the camera, is very difficult to model. The impact of unwanted camera motion on the observer is poorly understood, and there is still a lack of an established and widely accepted perceptual model to quantify the visual discomfort that may result from camera

movement or other electronic instabilities. Furthermore, to achieve an efficient video stabilization, one has to find a trade-off between the removal of unwanted camera motion and the introduction of artifacts, most notably losses in resolution and field of view. Indeed, another factor that limits the effectiveness of video stabilization is the appearance of an indefinite areas due to the fact that the motion zones may not completely match those of the field of view. The undefined zones need to be removed by cropping and filled-in using some image in-painting techniques [89] or other ad hoc solutions. While this can be used to remove distractions [9] and tighten the focus of the camera, it may also remove valuable areas from the scene. Finally, such evaluations depend strongly on the content of the captured scenes and the types of camera movements to be removed. This opens the field to a multitude of possible scenarios that is difficult to represent through a single reference database that could be used by the scientific community working on VS.

Moreover, in the absence of a well accepted methodology for comparing the existing video stability methods, it is difficult to effectively judge the impact on the quality of the output at each step of Video Stabilization. Indeed, video stabilization pipeline is composed of several successive steps, making thus the overall objective VSQA rather a hard task without the use of an effective VSQA metric taking into account the specificity of each step of the process. Such efficient VSQA metric would help in comparing the available VS techniques for a given application. While some parts of the stabilization process, such as the camera motion estimation, can be evaluated separately, some other components of the VS pipeline could not be evaluated independently. Specifically, the choice of how to stabilize the camera motion is dependent on the desired output. Hence, the main objective of this study is to provide a comprehensive overview and a framework for developing an effective methodology for VSQA. Both subjective VSQA protocols and objective evaluation through quantitative measures are considered.

3.2 Background

In this section, we propose to review the different attempts in the literature to assess the performance of video stabilization methods. Two main approaches are considered: subjective evaluation based on user studies and objective evaluation based on metrics.

3.2.1 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 methods for VSQA are often preferred. There are two ways to perform subjective evaluation. A user study can be conducted on a pool of observers to record their preferences between two or more stabilization methods. The alternative is to rely on video examples of that exhibit the performances of different stabilization methods on similar videos.

User studies

Formal user studies, due to their complexity, have rarely been used for VSQA. To our knowledge, only five documented studies exist in the literature.

- Koh & al. [11] test their stabilization method against commercial algorithms: the Youtube Stabilizer [4], and the Warp Stabilizer of Adobe After Effects [18]. They used a database containing the most challenging problems for video stabilization. This set of videos is organized into 7 categories: “Simple", “Object", “Depth", “Rolling Shutter", “Crowd", “Driving" and “Running", according to the video content and the challenges that it presents for video stabilization methods. The dataset contains a total of 162 videos assembled from various publications [4], [18], [29], [34], [76], [90] and videos available on the Internet. 50 users participated in the study. Each was shown 3 randomly selected videos from each category, for a total of 21 videos, using pairwise comparison between the method proposed by Koh & al. and one of the commercial stabilizers. The placement of the two videos was randomly determined for each video to avoid bias, and users had to choose between “ Method A ", “ Method B " or “ No Preference ". Thee users were instructed to neglect differences in aspect ratios, contrast or sharpness. Instead, they were told to focus on deformations of scene structures, rolling shutter distortions, and wobbling or shaking.
- Zhang & al. [9] also compare their method to the Youtube and Warp stabilizers, but use 3 different versions of their algorithm (simple stabilization, low-level optimization and high-level optimization). The study included 25 participants, 15 male and 10 females, aged 20 to 30, and used 16 videos collected from the internet. Subjects were shown the original video juxtaposed with each stabilized version, with the order randomly selected. They

were asked to rate the stabilized video on 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. The first three criteria were rated between -4 (important degradations compared to the original video) to +4 (important improvements on the original video), while the cropping was rated between -4 and 0, as it is impossible to obtain a better field of view than the original video.

- Liu & al. [76] use a dataset comprised of 174 videos between 10 and 60 seconds, divided into seven categories: “Simple", “Quick rotation", “Zooming", “Large parallax", “Driving", “Crowd" and “Running" . The study involved 40 participants, using four randomly chosen videos from each category to obtain 28 samples for each participant. The evaluation used pairwise comparisons between the proposed algorithm and either the Youtube Stabilizer or the Warp Stabilizer, but unlike previous instances, users were unable to choose “no preference". Users could play videos at the same time or one by one, pause or restart the video as they saw fit. They were told to ignore differences in ratio or sharpness, which could be caused by the codec used in the algorithms. Participants were asked after the test the criteria they used to make their choice.

- Wang & al. [31] compare their algorithm to that of Liu & al. [76] and Matsushita [30]. The study included 78 participants, using 10 video samples. Each participant was asked to give each stabilization result a score between 0 and 100 (the higher the score the more effective the stabilization). The different results were shown randomly and anonymously. Participants were asked to give accurate evaluations, and were allowed to watch each video several times before giving their score.

- Zhang & al. [88] created a full-reference dataset. Two cameras, an osmo camera equipped with a mechanical stabilizer, and a goPro camera without stabilizing equipment were set as close as possible, with similar resolution and frame-rate. Different scenes were filmed using the cameras, after which the videos were truncated to retain only the portions visible on both videos. This dataset contains 9 categories: "walking", "climbing", "running", "riding", "driving", "large parallax", "crowd", near-range object" and "dark", with five videos in each category. Because videos have been truncated, the resolution varies but lies around 940 pixels in width and 500 in height, while the duration averages to around 15 seconds per videos. Four stabilizers were tested: Adobe After Effects warp stabilizer, Google Youtube Stabilizer, Deshaker and the temporally optimized stabilization. Twenty participant rated each method on each video. Participants viewed two videos simultaneously: the video taken with a mechanical stabilizer and

the corresponding video after being processed by one of the aforementioned algorithm. They were unaware as to the stabilization process used for each video. Both videos were rated from 1 (best) to 5 (worst). The score of the digital stabilization method was the difference between the two ratings.

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 efficiently 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 algorithm.

Visual inspection

Besides the already mentioned limitations, user-studies are also time-consuming and difficult to set up. This is probably the main reason why they are not more present in the literature. However, there a still a need for subjective evaluation since it is the easiest and most complete way of assessing video stabilization [85].

To address this unsolvable problem, most authors invite the reader to perform this evaluation themselves. They present examples of their results on various datasets and leave the subjective evaluation up to the reader. These datasets usually encompass a variety of scenes, and include stress cases were the proposed method fails to highlight the limitations of the process. Comparisons with previous methods are often shown. Because the implementations of different methods are not always available, it is not always possible to obtain new results for previously proposed methods. It is therefore quite common to use previously given examples of such stabilization method, alongside the results of the newer stabilization method on this video, to compare two methods. Because these results are difficult to view on paper, it is frequent to add markers such as a cross targeting a specific feature to help visualizing the misalignment in the original video or in failed cases compared to successful stabilizations.

3.2.2 Objective evaluation

There are few objective measures that have been used for video-stabilization quality assessment. While some intermediate steps can be evaluated using ground-truths, such as the estimation of the camera path, other steps are much harder to quantify. The determination of the new camera path, for instance, relies both

heavily on the camera motion estimation and subjective ideas of what the movements of the camera should be. Meanwhile, the evaluation of the end result is challenging as it is based on several criteria, some of which are inherently subjective. So far, several metrics based on different criteria have been suggested but none have been widely accepted, and neither has strong correlation between the subjective preferences and objective metrics proposed been shown.

Several approaches have been used to define relevant metrics. Some use a blind setting and base their metric on the end result of the stabilization, i.e. on the stabilized video, that should exhibit good properties. Some only assess the estimation of the camera parameters, using ground truth data obtained via synthetic videos or controlled video capture.

Metrics without ground-truth

In most cases, no ground-truth (i.e. perfectly stabilized video) is available. Evaluation thus consists in either comparing the stabilized video to the original unstable video or to judge the quality of the stabilized video in itself. Thus, evaluating the end result of video stabilization is difficult because several criteria come into play.

The most common evaluation metric is the Inter-frame Transformation Fidelity (ITF) index [91], [92], based on the inter-frame PSNR. This measures the similarity between successive frames, which is assumed to be higher when the camera path is smooth. In this context, good video stabilization should produce a video with a larger ITF. However, moving objects will negatively impact the PSNR without necessarily affecting the stabilization.

Several metrics have been proposed to evaluate the smoothness of the camera motion using the detected movements in the video after stabilization. The average of the velocity of pixels between adjacent frames is one of the ways to estimate the camera instability [31]. However, since the most disturbing factor in the camera movements are not fast camera motion but abrupt changes in the camera motion, the acceleration can be used instead [85], specifically the acceleration observed in feature trajectories. Dong et al. [14] use a closely related metric that measures the differences between the accelerations in the original video and the smoothed video.

Moreover, video stabilization introduces artifacts, which must be taken into account when evaluating the results. Since the stabilization process introduces a reduction in resolution to avoid undefined ares, this provides an intuitive way to evaluate the losses caused by the stabilization. Two approaches exist: the percentage of undefined area before cropping gives a certain idea of the information

loss due to the camera rectification [13], [39], while the cropping ratio informs us to the resolution loss after the cropping scheme used [76]. Another artifact introduced by certain methods are distortions caused by perceptual motion models or certain warping schemes. Such distortions are often specific to certain methods, which define distortions differently [76], [79].

Metrics based on ground-truth

Several approaches have been proposed to evaluate the camera motion modeling, as it is both a key component of video stabilization and one of the components for which it is possible to obtain a ground truth. While for most videos such ground truth is inaccessible, using synthetic video with digital tools such as Blender or Maya Autodesk allows the comparison between the known original 3D path of the camera, and the modelled camera path. Using the difference between each estimated camera motion parameter and the ground truth, both the average error [79] and the maximum error can be used to evaluate the camera motion estimation [8]. The maximum error is important as a large error, even for a single frame, can induce drastic cropping, while the average error give a good idea of the motion estimation accuracy. For methods using 2D motion models, 3D synthetic scenes are unsuitable, instead still video are captured using mechanical stabilizers such as tripods and artificial transforms are applied to simulate video shaking with known 2D transform parameters. A common metric used is the Root Mean Square Error [43], [44], that measure directly the differences between the expected and observed position of pixels. Since different parameters can have very different values yet similar impacts, this measure can be easier to interpret.

In addition to completely synthetic sequences, it is possible to control the scene so as to obtain pseudo-ground truth. Jia et al. [7] captured videos using a smartphone lay-ed out on the ground, rotating it, then replacing it in the original position. This constrains the motion to mostly a rotation along the z-axis, and ensures that the final angle of the camera should be identical to the beginning. The difference between the estimated parameters for the z-axis of the camera at the start and end of the video can give an estimation of the camera motion accuracy. Aguilar et al. [56] test their method on videos captured by A.R.Drones 1.0, but they also set up a camera on a tripod to film the aforementioned drone to obtain its motion in the camera plane, which can be compared to the estimated motion of the drone. Methods using feature tracking can use the feature positions to evaluate the camera motion estimation. After determining the camera motion between two adjacent frames t and $t+1$, the projected positions of feature points of t after applying the estimated camera motion can be compared to the known

positions of the features at frame $t+1$. Jeon et al. [12] uses different features and compare errors obtained between the projected and observed feature positions.

Finally, [88] use two cameras, one with mechanical stabilizer and one without, to obtain videos with full reference. Using these references, they define a metric based on the distance between the camera path of the digitally stabilized video with the mechanically stabilized video. To obtain accurate path while avoiding complex structure-from-motion, the videos are segmented by planes and a separate 2D path is determined for each plane (using homographies).

3.3 Proposed framework

The aim of this section is to investigate the performances of several video stabilization methods, on the same database, with the same objective metrics. These results will be confronted with subjective assessment by several viewers so as to better understand the relevance of the objective metrics and their link with actual subjective perceptions.

Overall, this study is based on

- 35 videos reflecting different challenges of video stabilization (see Section 3.3.1)
- 4 standard video stabilization methods (see Section 3.3.2)
- 5 standard metrics for VSQA and 2 image/video metrics (see Section 3.3.3)
- 18 viewers and a subjective evaluation framework (see Section 3.3.4)

3.3.1 Database

The performance of four representative video stabilization methods is evaluated using the database assembled by Koh et al. [11] from different publications [4], [18], [29], [34], [76], [90]. This database contains the most challenging problems for video stabilization. This set of videos is organized into 7 categories: “Simple", “Object", “Depth", “Rolling Shutter", “Crowd", “Driving" and “Running", according to the video content and the challenges that it presents for video stabilization methods. Sample thumbnails of the videos are displayed on Figure 3.1.

Videos from the “Simple" category contain smooth depth differences and slow camera motion, which do not present any particular challenge to most video stabilization methods. Videos from the “Object" category contain large moving

objects in the foreground, which represent a very challenging problem as it is very difficult to discriminate between camera-induced motion and motion resulting from moving objects. Therefore, any common VS method that could not discriminate between these movements may produce inconsistent results. Videos from the “Depth" category contain strong depth discontinuities. This is another very challenging problem for VS methods based on 2D motion models since the depth aspect is not properly taken into account in these approaches and especially in the case of important depth. The “Rolling Shutter" category consists of videos taken using a camera that captures the video row by row. In the case of fast camera motion this may produce noticeable video distortion due this scan-line acquisition system. The videos in this category contain such artifacts, usually caused by fast lateral motion which tilts vertical structures of the scene. Videos from the category “Crowd" contain many independent movements and occlusions, which makes the process of determining the camera motion to stabilize more complicated. The “Driving" sequences are taken from videos embarked on moving vehicles. These videos contain a main steady forward motion disturbed by high frequency shakes of variable intensity, as well as important depth differences and occlusions or moving objects. The "Running" sequences contain excessive shakes that are difficult to correct while maintaining a good video resolution as well as important depth differences.

3.3.2 Methods

In this work, we propose to assess the performances of four standard video stabilization methods. These methods have been chosen because they are representative of the current approaches that appear in the literature. They are also all free of charge and their source-code or software are available, which allows to conduct a fair comparison of the considered VS methods. In the following, we provide a brief description of the VS techniques used in this study.

- Deshaker [93] is a free plugin that can be used within the VirtualDub software. It is a fast, free and ready-to-use tool that stabilizes horizontal/vertical panning, rotation and zooming. The method assumes that the camera movement between two successive frames can be modelled as a 2D transform (homography, affinity...). Its interface offers a large number of settings and parameters that can be useful for advanced users. In the following experimental setting, we have used the default parameters of these softwares. It uses a block-matching algorithm to determine movements within the video with a coarse-to-fine approach to handle large movements. Default settings uses 30-pixel blocks, 30% of the coarser frame as the initial

search range then 4 pixels for the search refinement at each iterations, ending with the frame at half the original scale. Motion vectors are discarded if either the maximum pixel difference or the combined differences between blocks are above given thresholds respectively 20 and 300), or if a pixels move too far in a direction that does not fit with the camera model (with a threshold of 4 pixels). The correction applied is computed by minimizing the squared correction and the motion acceleration, with a maximum correction for both panning, zoom and rotation (set to 15 degrees for panning, 15% for zoom and 5 degrees for the rotation). It also allows to consider rolling shutter effect when determining and correcting the camera motion.

- Youtube Stabilizer [4], [94] is arguably the most popular video stabilization software. Approximately 300 hours of videos are uploaded in Youtube each minute, and the Youtube Stabilizer is a routine discretionary option in the upload process. Similarly to Deshaker, the method assumes a simple 2D geometrical transform between two successive frames, but includes a L1-regularization in the smoothing process, that mimics the motions produced by professional cameramen. The strength of the stabilization is finely tuned so as to ensure that the region or subject-of-interest is always visible in the stabilized video. It also uses homography mixtures to detect and correct rolling shutter artifacts.

- Sanchez et al. method [82] is a recent video stabilization method, publicly available on IPOL [13]. This method assumes a 2D homography transformation between two successive frames involving 8 parameters. It is based on a local smoothing of the transform parameters used in the computation of the stabilized video.

- Koh et al.method [11] is also a recent method which, unlike other VS approaches, does not assume any 2D transformation between consecutive frames. Instead, this approach attempts to provide a plausible and perceptually satisfying correction, that is not based on the geometrical reality of the scene. This method also includes an explicit rolling shutter removal step, and is able to handle large objects in the foreground.

3.3.3 Metrics

Here, we focus on five common VSQA metrics, namely : Inter-frame Transformation Fidelity (ITF), Average Speed (AvSpeed), Average Acceleration (AvAcc), Average Percentage of Conserved Pixels (AvPCP) and Inter-frame Similarity Index (ISI), based on the structural similarity index(SSIM). We also used two metrics

used for image and video quality evaluation : Spatial Efficient Entropic Differencing for Quality Assessment (SPEED-QA)[95], and a blind video quality metric known as VIIDEO [96].

Interframe Transformation Fidelity (ITF)

The most widely used metric for assessing the performance of video stabilization methods is the Interframe Transformation Fidelity (ITF) index [91], [92]. It is based on the video inter-frame PSNR. Given a video I composed of N_f frames, ITF is expressed as the average inter-frame PSNR.

$$\text{ITF} = \frac{1}{N_f - 1} \sum_{i=1}^{N_f - 1} \text{PSNR}(t), \tag{3.1}$$

where PSNR(t) is the peak signal-to-noise ratio (in dB) based on the mean-square-error between frames $I(t)$ and $I(t+1)$. The intuitive idea behind this metric is that, if the camera movement is smooth (i.e., stabilized video), the similarity between the consecutive frames should be larger than in the presence of strong camera motion. This metric can also assess the distortions and photometric artifacts that may result from the stabilization process. Note that, if no objects/subjects are in movement in the video and if the stabilization is perfect, the ITF would tend towards infinity.

Interframe Similarity Index (ISI)

Another metric that can be used in a similar way is the Structural Similarity Index (SSIM) [97]. In order to extend its application to video streams, we define the Interframe Similarity Index (ISI) as the average of the SSIM between successive frames across the video. This new VSQA metric is given by:

$$\text{ISI} = \frac{1}{N_f - 1} \sum_{i=1}^{N_f - 1} \text{SSIM}(t), \tag{3.2}$$

where SSIM(t) is the structural similarity index between frames $I(t)$ and $I(t+1)$. High values of ISI mean that successive frames are perceptually similar, which is provide better visual comfort for viewer.

Average Speed (AvSpeed)

Another aspect that could be analyzed when judging the quality of the stabilized video is the local motion. Indeed, the stabilization process acts on the movements present in the video. The stabilization process should produce a smoothing effect of these annoying movements leading to a fluid video sequence. This can be checked by extracting salient feature points in the video and analyzing their displacement along the video. Intuitively, in a properly stabilized video, the movements of these feature points are smooth, i.e. with small speed/acceleration. Considering the i^{th} features point with coordinates $z_i(t)$ in frame $I(t)$, its movement can be characterized by its instantaneous speed $\dot{z}_i(t) = z_i(t+1) - z_i(t)$ or its instantaneous acceleration $\ddot{z}_i(t) = z_i(t+1) - 2z_i(t) + z_i(t-1)$. The Average Speed (AvSpeed) metric is expressed as the average speed of all feature points along the video [31]. If a total of N_p feature points are extracted in the video, the AvSpeed metrics is given by:

$$\text{AvSpeed} = \frac{1}{N_p(N_f - 1)} \sum_{i=1}^{N_p} \sum_{t=1}^{N_f - 1} \left\| \dot{z}_i(t) \right\|_2 , \tag{3.3}$$

and is defined as the average quantity of movement of the feature points: it should be as low as possible. In the literature, several feature points have been proposed for motion computation (SURF, SIFT, KLT, sparse optical flow...). In our experiment, we make use of the KLT descriptors thanks to their relatively low computational cost and efficiency.

Average Acceleration (AvAcc)

The analysis of the quantity of movements present in the video may not be sufficient to assess the qualitative aspects of video stabilization. Indeed, the perception of the movement is not only linked to the quantity but also to the type of movement present in the video. In particular, professional cameramen use hardware solutions (such as steadycam), that tend to produce movements with linear displacement or speed [71]. Furthermore, if the cameraman is attempting to follow a subject of interest, the observed motion is intentional and should not be removed. For these reasons, several authors have considered that stabilization should be assessed according to the acceleration rather than the speed of feature points. The Average Acceleration (AvAcc) metric [79] could be expressed as:

$$\text{AvAcc} = \frac{1}{N_p(N_f - 2)} \sum_{i=1}^{N_p} \sum_{t=2}^{N_f - 1} \left\| \ddot{z}_i(t) \right\|_2 , \tag{3.4}$$

This metric quantifies the average acceleration of the feature points. For a good stabilized video, this measure is low.

Average Percentage of Conserved Pixels (AvPCP)

Video stabilization naturally induces a resolution loss in the processed video. Indeed, in case of strong stabilization, the inverse 2D or 3D transform applied on frames often create unknown area in the video that cannot be interpolated without additional information. To circumvent this limitation, one has to apply some post-processing solutions such as cropping, zooming or video re-sizing, so as to remove those blank areas. The loss of resolution, if large, may produce an annoying effect and can therefore be considered as a criterion of evaluation [82] . One way to quantify this effect is to express the ratio of pixels that survive the VS process. Given the original video I and a stabilized video $\tilde{I}$, the Average Percentage of Conserved Pixels (AvPCP) could expressed as the following ratio:

$$\text{AvPCP} = \frac{100}{N_f} \sum_{t=1}^{N_f} \frac{\text{res}(\tilde{I}_t)}{\text{res}(I_t)}, \tag{3.5}$$

where res(.) is the resolution (in pixels) (i.e.res$(I_t) = MN$ if I_t if of size $N \times M$). Obviously, a well stabilized video should have high AvPCP value so as to minimize the fraction of lost pixels during the stabilization process. It is worth noticing that the interpretation of this metric greatly depends on the level of stabilization: in particular, note that this metric equals 100% if no stabilization is performed. This metric should only be used in addition to other performance metrics or to evaluate methods with the same level of stabilization.

SpEED-QA

Spatial Efficient Entropic Differencing for Quality Assessment (SpEED-QA) [95] is an image quality metric that uses natural scene statistics (NSS) to determine the quality of an image. It is based on the observation that an image treated with local mean removal should be close to a gaussian scale mixture. This criterion is used on both the original and distorted image, and the difference is used to evaluate the loss of quality, with smaller scores indicating smaller losses. To evaluate an image, SpEED first computes the locally mean subtracted images

I_{lms} for both the reference and distorted image:

$$I_{lms}(z_t) = I(z_t) - \mu(z_t), \tag{3.6}$$

$$\mu(x_t, y_t, 1) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} I(x_t + k, y_t + l, 1) \tag{3.7}$$

In this equation, $w_{k,l}$ is a Gaussian weight. The resulting images are divided into M blocks to obtain local assessments of the degradations: C_m is the m^{th} block of I_{lms}. Since this is applied to both the reference and distorted images, C_{mr} denotes the block of the reference image while C_{md} denotes the block of the distorted image. These blocks are modelled as Gaussian scale mixtures (GSM):

$$C'_{mr} = S_{mr} U_{mr} + W_{mr}, \tag{3.8}$$

$$C'_{md} = S_{md} U_{md} + W_{md} \tag{3.9}$$

where S_{mr} and S_{md} are non-negative random variables representing the salience of the block, while U_{mr} and U_{md} are Gaussian random vectors and W_{mr} and W_{md} are Gaussian noise. Observations of NSS indicate this model should hold best for pristine videos. The quality of the block can therefore be evaluated using the conditional entropies $h(C'_{mr}|S_r = s_r)$ and $h(C'_{md}|S_d = s_{md})$. To give greater importance to salient areas, the entropies are locally weighted by the scalar factors γ_r and γ_d :

$$a_{mr} = \gamma_r mr(C'_{mr}|S_{mr} = s_{mr}), \tag{3.10}$$

$$a_{md} = \gamma_{md} h(C'_{md}|S_{md} = s_{md}) \tag{3.11}$$

where $\gamma_{mr} = \log(1 + s^2_{mr})$ and $\gamma_{md} = \log(1 + s^2_{md})$. The final SPEED score is obtained using the differences between the reference and distorted entropies, and averaging the results for each block:

$$\text{SPEED-QA} = \sum_{m=1}^{M} |a_{mr} - a_{md}| \tag{3.12}$$

This metric indicates whether the distorted image exhibits statistical anomalies more pronounced than the reference image. It has been used both as an image quality metric and to evaluate video quality [98] by measuring the differences between adjacent frames, and has been shown to perform as well or better than SSIM or PSNR on video and image quality databases such as LIVE [99], [100] VQA.

VIIDEO metric

VIIDEO [96] is a no-reference video quality metric that measures the correlation between scene statistics in the video and a filtered version of the video over time. Rather than work on video frames directly, VIIDEO uses the differences ΔI_t between adjacent frames t and $t+1$, to exploit the temporal changes in the video.

$$\Delta I_t = I_{t+1} - I_t, \forall t \in \{1, 3, 5, \ldots, \frac{N}{2}\} \tag{3.13}$$

These frame differences are compared to a the filtered differences ΔG_t using a Gaussian filter in the spatial domain:

$$\Delta G_t(x, y) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} \Delta I_t(x + k, y + l) \tag{3.14}$$

where $w_{k,l}$ is a Gaussian weighting function. Both the initial and filtered differences are then treated using local mean removal and contrast normalization:

$$\Delta \hat{I}_t(z_t) = \frac{\Delta I_t(z_t) - \mu_t(z_t)}{\sigma_t(z_t) + C} \tag{3.15}$$

$$\Delta \hat{G}_t(z_t) = \frac{\Delta I_t(z_t) - \mu_t^g(z_t)}{\sigma_t^g(z_t) + C} \tag{3.16}$$

where $\mu_t(z_t)$ is the local mean of ΔI_t and $\sigma_t(z_t)$ the local standard deviation of ΔI_t, while $\mu_t^g(z_t)$ and $\sigma_t^g(z_t)$ denote the local mean and standard deviation of ΔG_t. A property of such normalized differences is that adjacent coefficients show regular structures that are disturbed by distortions. To exploit this, the products of adjacent coefficients in vertical, horizontal and diagonal directions are computed. Such products can be asymmetric generalized Gaussian distributions. The parameters of these distributions are used as descriptors, respectively Φ_t and Γ_t. The temporal differences of these vectors are captured over a time-frame of length S, noted :

$$A_t(x, y) = \{\Phi_{t+1+s}(x, y) - \Phi_{t+s}(x, y), \forall s \in \{1, 2, ...S\}\} \tag{3.17}$$

$$B_t(x, y) = \{\Gamma_{t+1+s}(x, y) - \Gamma_{t+s}(x, y), \forall s \in \{1, 2, ...S\}\} \tag{3.18}$$

Coefficient $\theta^{t+\nu k}$ are then defined as the coefficients of the co-variance between $A_{t+\nu k}(x, y)$ and $B_{t+\nu k}(x, y)$, representing the correlations between the structural changes in the original and filtered video.

$$\text{VIIDEO} = \sum_{t+\nu k} \sum_{f} \theta_f^{t+\nu k} \tag{3.19}$$

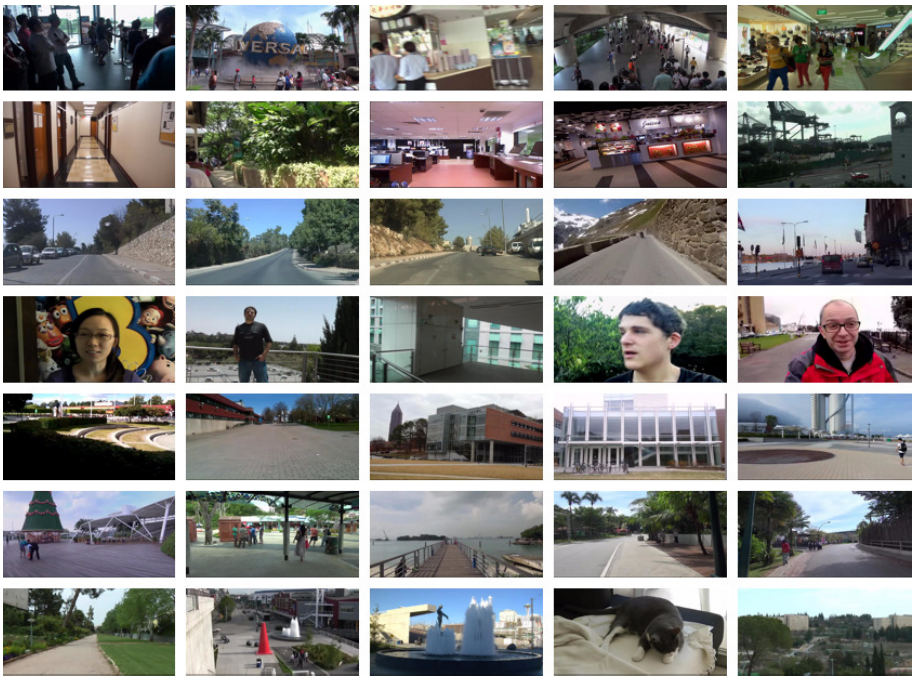

Figure 3.1 Sample thumbnails of the videos used in this study.

Lower scores indicate greater differences between the initial and filtered video. Since filtering has a greater impact on high quality videos, high quality videos have lower scores than low quality videos. It has been used to evaluate the effects of compression on video quality [101] It is particularly interesting because unlike several no-reference quality metrics, it is not trained on any dataset, meaning that it is not tailored to a specific type of scene, which is valuable to assess video stabilization as the effects and challenges vary greatly for different types of scenes.

3.3.4 Experimental setup

We use pairwise comparisons to determine the preferences between different video stabilization techniques. We selected a dataset comprised of 5 videos from each of the 7 categories in the previously mentioned database in order to perform visual tests, for a total of 35 videos. Sample images are shown in figure 3.1 In order to facilitate the visual tests, shorter videos were chosen and truncated to 10 seconds short when they exceed this duration. The 35 videos from the dataset were treated with the stabilization algorithms described above. The original videos

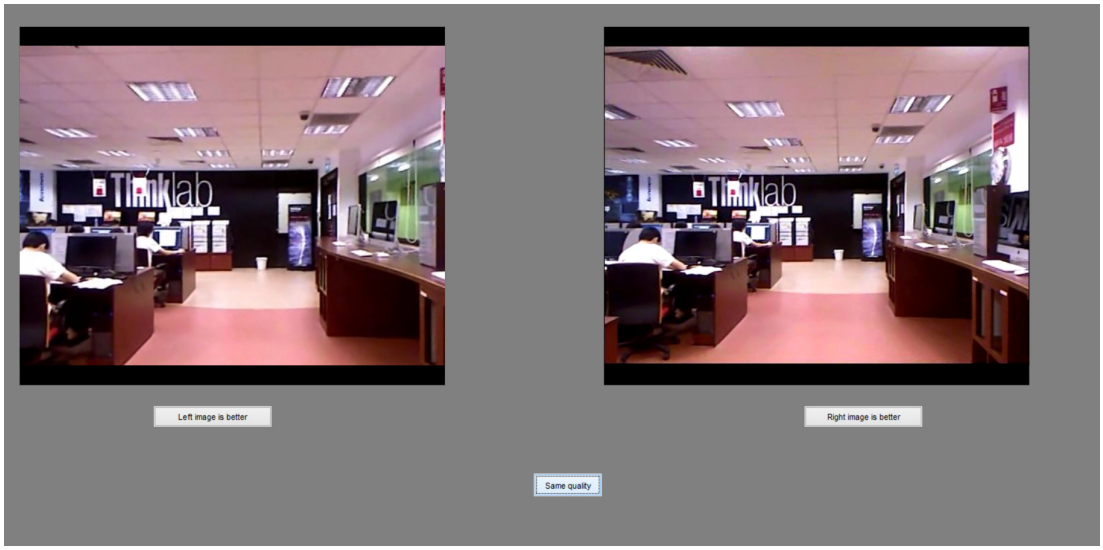

Figure 3.2 Exemple of the evaluation setup.

were also included to account for the side effect of the VS methods that may degrade the input videos, resulting in 5 versions of each video. Observers were shown two different versions of the same video, and had to choose which one they preferred, according to their own criteria. They also had the option to indicate no preference between the two. Each observer was shown 50 randomly selected pairs of videos, with each of the 35 videos shown at least once. Each pair consists on two randomly chosen versions the video. Videos could be replayed once, but users were instructed that they could chose their preference-choice at any time. In doing so, the negative effects of visual fatigue and the reduction of visual attention are minimized. The tests were performed on eighteen observers at the Laboratoire de Traitement et Transport de l'Information(L2TI) on a calibrated LCD monitor in a controlled environment as the one used for image quality assessment and described in [102]. The observers were students and members of the laboratory, most of which were not experts in video stabilization or quality. The average age was 31 years old.

3.4 Results and discussion

The performance evaluation of the 4 VS methods has been done on a set of 7 video categories representing various challenging scenarios as mentioned previously. In the following we provide a brief discussion on some preliminary results by considering both subjective and objective aspects.

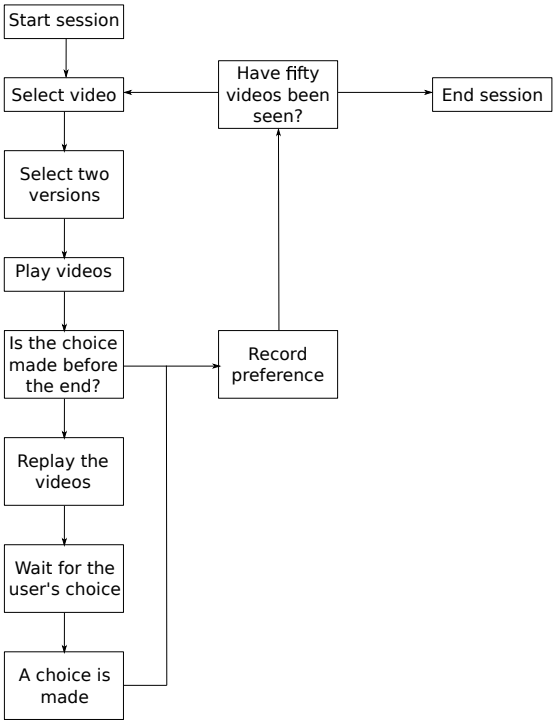

Figure 3.3 Workflow of the evaluation protocol

3.4.1 Objective performance evaluation

Tables 3.1 and 3.2 display the values of the objective metrics obtained by the methods and video categories described in Section 3.3. The first observation is that the method by Koh et al. obtains the best general performances according to the AvSpeed, AvAcc and VIIDEO criteria, and second best on the ITF and ISI metrics. It outperforms other methods on all categories for at least one metric of evaluation. Whereas, Deshaker obtains the lowest performances for both ITF and ISI, but the best result according to the AvPCP and SpEED-QA metric. Using the default settings, this method specifically avoids excessive corrections in order to maintain video resolution. As expected and described in Section 3.2.2, the AvPCP is in contradiction with other metrics: there is a compromise to make between keeping the video resolution high and perform a severe stabilization. Interestingly, the metric values are sometimes better on the original unstabilized video than on the stabilized one. This phenomenon is especially observed with Youtube and Sanchez et al. methods, for instance on the Driving or Crowd categories. In fact, since these methods are based on 2D geometrical models, they sometimes fail at dealing with videos in which the camera movement lies in a 3D space or includes parallax effects. This causes erratic stabilizations that mis-estimate the camera movement and actually create additional distortions in

		original	Deshaker [93]	Youtube [4]	Sanchez et al. [82]	Koh et al. [11]
crowd	*ITF*	19.03	20.35	18.24†	**22.67**	22.42
	AvSpeed	3.83	2.71	5.78†	2.81	**2.52**
	AvAcc	1.12	0.40	0.90	0.43	**0.33**
	AvPCP	100	**78**†	69†	54†	74†
	ISI	0.62	0.70	0.60†	**0.78**	0.72
	SpEED-QA	0.51	**0.49**	0.54†	0.52†	0.50
	VIIDEO	0.13	0.11	0.17†	0.10	**0.09**
depth	*ITF*	21.67	22.05	22.7	23.8	**24.01**
	AvSpeed	3.50	2.55	2.99	2.65	**2.21**
	AvAcc	2.84	0.79	2.20	0.58	**0.53**
	AvPCP	100	83†	**92**†	64†	83†
	ISI	0.70	0.76	0.75	0.80	**0.82**
	SpEED-QA	0.54	0.49	0.56†	**0.45**	0.49
	VIIDEO	0.11	0.09	0.10†	**0.08**	0.08
driving	*ITF*	23.18	20.88†	22.03†	22.78†	**23.11**†
	AvSpeed	2.56	1.39	1.36	1.42	**1.23**
	AvAcc	2.85	0.86	0.94	0.91	**0.53**
	AvPCP	100	**85**†	79†	62†	78†
	ISI	0.69	0.69	0.75	0.75	**0.76**
	SpEED-QA	0.76	**0.47**	0.61†	0.53	0.56
	VIIDEO	0.11	0.10	0.10†	**0.09**	0.09
object	*ITF*	21.17	22.64	23.21	**24.95**	23.88
	AvSpeed	3.54	1.99	3.07	3.81†	**1.89**
	AvAcc	2.21	0.93	0.88	1.38	**0.64**
	AvPCP	100	**80**†	70†	49†	79†
	ISI	0.68	0.76	0.76	**0.85**	0.79
	SpEED-QA	0.62	0.50	0.54	**0.49**	0.56
	VIIDEO	0.12	0.10	0.12	0.12	**0.10**

Table 3.1 Results on the objective metrics (part 1). The scores denoted with † correspond to cases where the stabilized video has worse objective quality level than that of the original video.

the video, which are hard to estimate as those metrics show good results on image similarity metrics. Apart from the AvPCP metrics, it appears that all metrics are coherent with each other: although their ranking is somewhat different. Globally, it could be noticed that the metric values are of the same order for the considered dataset.

		original	Deshaker [93]	Youtube [4]	Sanchez et al. [82]	Koh et al. [11]
rollingShutter	*ITF*	20.65	24.09	**29.75**	28.62	26.43
	AvSpeed	7.29	2.09	2.32	**1.46**	1.70
	AvAcc	6.16	1.15	0.50	0.90	**0.40**
	AvPCP	100	**69**†	**69**†	60†	63†
	ISI	0.59	0.77	**0.88**	0.86	0.86
	SpEED-QA	0.57	0.53	0.61†	0.56	**0.56**
	VIIDEO	0.15	0.09	**0.06**	0.10	0.09
running	*ITF*	18.46	22.9	22.65	**26.57**	25.88
	AvSpeed	7.14	1.91	3.32	1.74	**1.67**
	AvAcc	3.14	0.74	0.96	0.78	**0.41**
	AvPCP	100	**61**†	52†	37†	57†
	ISI	0.56	0.69	0.76	**0.85**	0.75
	SpEED-QA	**0.51**	0.52†	0.56†	0.53†	0.56†
	VIIDEO	0.16	0.08	0.12	0.08	**0.08**
simple	*ITF*	24.48	27.87	**31.64**	31.33	28.64
	AvSpeed	3.24	1.29	**0.96**	1.06	1.34
	AvAcc	2.89	0.61	0.40	0.48	**0.36**
	AvPCP	100	**83**†	72†	72†	**83**†
	ISI	0.74	0.88	**0.93**	0.93	0.88
	SpEED-QA	0.60	0.57	0.57	**0.49**	0.57
	VIIDEO	0.16	0.08	0.12	0.08	**0.08**
average	*ITF*	21.23	22.97	24.32	**25.82**	24.91
	AvSpeed	4.44	1.99	2.83	2.14	**1.79**
	AvAcc	3.24	1.29	0.97	0.78	**0.46**
	AvPCP	100	**77**†	72†	57†	**74**†
	ISI	0.65	0.75	0.78	**0.83**	0.79
	SpEED-QA	0.59	**0.51**	0.57†	0.51	0.54
	VIIDEO	0.13	0.09	0.10	0.10	**0.09**

Table 3.2 Results on the objective metrics (part 2). The scores denoted with † correspond to cases where the stabilized video has worse objective quality level than that of the original video.

3.4.2 Subjective performance evaluation

Table 3.3 presents the sample preference matrix of the subjective pair comparison tests, averaged on all videos. The method by Koh et al. obtains the best overall results : it is preferred to any other methods in two-thirds of cases. This result is coherent with the results obtained with the IFT, AvSpeed and AvAcc metrics. The second best method is Deshaker, which may appear surprising according to the results of Table 3.1. In fact, the fact that Deshaker does not perform

	M1	**M2**	**M3**	**M4**	**M5**
M1	-	5.5	7	9	4.5
M2	12.5	-	11	10	7.5
M3	11	7	-	8.5	7.5
M4	9	8	9.5	-	6
M5	13.5	10.5	10.5	12	-

Table 3.3 Sample preference matrix aggregated on all videos for 18 observers. Score in case (i, j) corresponds to the average number of observers that preferred method i over method j. M1 = no stabilization, M2 = Deshaker [93], M3 = Youtube [4], M4 = Sanchez et al. [82], M5 = Koh et al. [11]

.

Metric	ITF	AvSpeed	AvAcc	AvPCP	ISI	SPEED-QA	VIIDEO
Kendall Rank Order Coefficient	0.12	-0.13	-0.23	-0.05	0.13	-0.07	-0.17

Table 3.4 Kendall Rank Order coefficient for the different metrics used

strong stabilizations and keeps a reasonable video resolution has for consequence that this methods never produce aberrant results, on the contrary to Youtube or Sanchez et al. Thus, strong stabilization is not the only criterion seeked by the observer, that prefers a video with acceptable camera movements than a fully stabilized video with distortions. The two last methods are Youtube and Sanchez et al. which perform a good stabilization (see Tables 3.1 and 3.2) with high image similarity metrics but fail on several videos and tend to produce over-cropping. Interestingly, no stabilization is sometimes preferred to any stabilization, which confirms that video stabilization is a processing step that can in fact cause distortions. It should be noted that both Youtube and Sanchez et al. seem to perform well in the “simple" category, and that as this database is representative of the range of challenges for video stabilization, it may not be representative of the usual type of scenes for which these methods are optimized.

3.4.3 Correlations between subjective and objective metrics

In Table 3.4, we computed the Kendall Rank Order coefficient [103] (KC) between the subjective assessment and the different objective metrics. This coefficient compares two ordered lists, for which each pair of elements is rated either concordant if the two elements are in the same order in each list, or discordant if they are ordered differently in both lists. The coefficient is then calculated using the difference between the number of concordant pairs CP and the number of discordant pairs DP normalized by the maximum value possible, which corresponds to the total number of pairs TP. This results in a coefficient between -1 and 1, where 1 indicates that all pairs are similarly ordered and -1 that all pairs are ordered differently in the two lists. Values closer to 0 indicate lesser or no correlation between the orders of the considered lists.

$$KC = \frac{CP - DP}{TP} \tag{3.20}$$

Here, we apply this between the results of the subjective evaluation and those of the objective metrics. For each objective metric, we look at every pair of videos evaluated subjectively and consider the pair concordant if the user preferred the video with the highest score, and discordant otherwise. This gives us a total of 750 pairs of videos to evaluate each metric.

The ITF and ISI correlate positively with the subjective evaluations while the SpEED-QA metric correlates negatively as expected, since higher scores indicate lower quality images. However, the low values for the coefficients indicate that such image quality evaluations are not a very good indicator of the effectiveness of video stabilization algorithms. The VIIDEO metric also correlates negatively as expected, but the higher score indicates that the temporal distortions evaluated are a better indicator of the subjective preferences compared to the image quality evaluation. Both the average speed AvSpeed and the average acceleration AvAcc are correlated negatively, as users prefer stable videos with fewer movements, but it is worth noting that the acceleration metric correlates much better than the velocity metric, indicating that it is the shifts in camera motion rather than the direct camera motion which is discomforting to users. Finally, the cropping ratio is rated negatively and extremely poorly, indicating that the loss of resolution is considered less important than the quality of the video or its motion. The low scores observed generally seem to reinforce the idea that multiple criteria affect video stabilization, and in particular it should be noted that the cropping ratio is often in opposition to other metrics, with aggressive stabilization obtain high values on most metrics but strongly reducing the video resolution.

3.5 Conclusion

Through this study it has been shown that the performance evaluation of video stabilization methods is far from being an easy subject. The few objective quality assessment measures that have been proposed in the literature do not include any knowledge of the effect of video instabilities on the human visual system. This study was not intended to propose such models but to encourage people to work in this direction and propose models to account for visual discomfort that may result from video instabilities. At the time of this study there have been no complete subjective and objective studies dedicated to VSQA. This contribution is a first step into the direction of filling this gap by proposing a video stabilization quality assessment methodology.

Chapter 4
Feature trajectories selection for video stabilization

Chapter 4

Feature trajectories selection for video stabilization

4.1 Introduction and motivations

As seen in Chapter 2, video stabilization operates in several interdependent steps, summarized on Figure 4.1. First, the video motion field is estimated using a frame-to-frame matching process. This estimation can be performed by tracking a set of salient features or points of interest in the successive frames. Several feature points trackers have been proposed in the literature. The most popular are SIFT, SURF or KLT [104]. The position of a given feature point throughout the video forms a feature trajectory, that represents the movement of an object in the video. Secondly, the original moving camera path is computed by using the estimated two-dimensional flow field. Usually a 2D or 3D motion model is used and the camera parameters are computed by solving linear equations. The camera path is then corrected and smoothed to obtain more coherent and smooth movement. Finally, a video restoration process based on the estimated camera path is used.

The estimation of the 2D or 3D camera parameters from feature trajectories is a tricky process since not all movements present in the video give information on the camera motion. While static objects are only affected by camera-induced movements, other objects undergo displacements that are caused by both the camera motion and the movements of the object in the scene. These moving objects need to be separated from the others and removed in order to compute the correct camera path. This task, referred to as *outlier removal* in this manuscript, is crucial, since it is linked to the ability of the method to deal with complex scenes containing several objects or subjects.

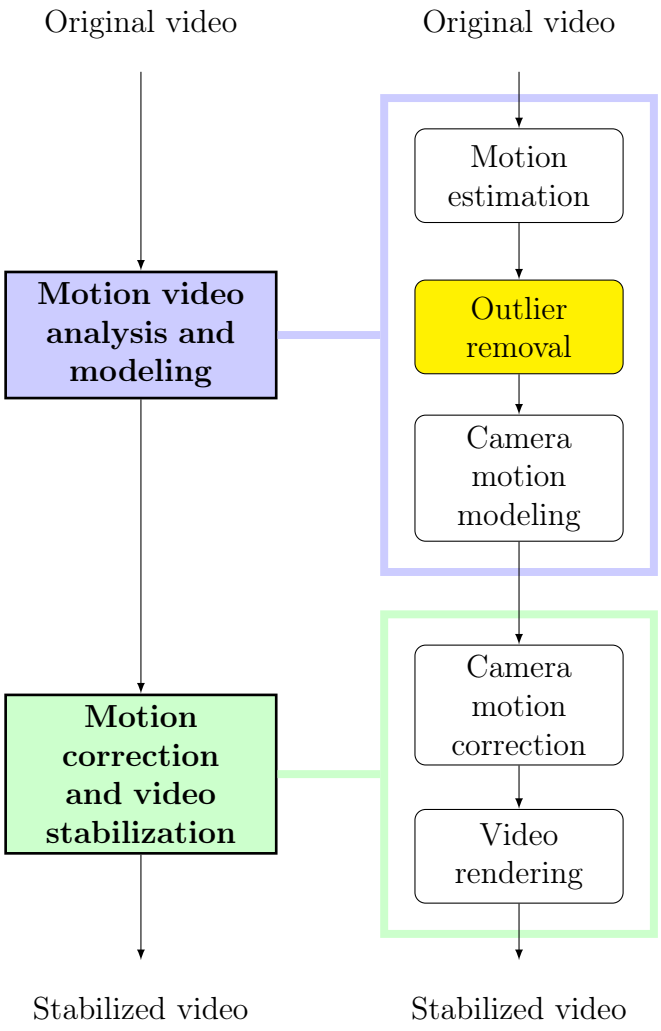

Figure 4.1 Main steps for video stabilization. The current chapter focuses on the outlier removal step.

4.1.1 Background

Interestingly, this step is often poorly documented in the papers and most of the authors do not address this issue or use ad hoc procedures. A complete overview of the state-of-the-art methods for outlier removal is presented in Section 2.2. For the sake of completeness and clarity, a brief overview of the main approaches is provided in the following.

The most common approach to handle this issue is based on the RANSAC algorithm [105] which is a parametric motion model that aims to quantify the point-to-point geometric correspondence between two successive frames (e.g. affine transform, homography...)[34], [106]. The feature points that do not fit the model are considered as outliers and removed. In RANSAC, outliers are detected by thresholding the projection error, but other approaches using the number of false alarms or negative log-likelihood can avoid the problem of fixing such a threshold [53]. By assuming low camera movements, simple strategies can be implemented by removing feature points with a velocity above a given threshold [45]. Alternatively, neighborhood information could be exploited to remove undesirable moving objects under the assumption of locally smooth motion vector field. The dense optical flow assumption could be then used to remove spurious movements by thresholding the motion gradient [107]. Delaunay triangulation can be used to establish neighborhood constraints, removing points whose motion differs from those of points lying along an edge of the triangulation [108].

4.1.2 Limits of the RANSAC approach

While all these strategies are efficient on simple cases where the vast majority of feature points belong to static objects or background, they provide poor results when the global assumptions they are based on are not valid (camera movements that fit a parametric model, low amplitude and spatially coherent movements). In particular, large objects moving in the foreground or scenes with many moving objects can prove difficult to handle. Furthermore, all these methods reject feature points based on the observation of two successive frames. They do not consider the feature trajectory during its whole lifetime. For instance, the same feature point may be considered as an inlier for certain pair of frames and as an outlier elsewhere. As such, the movement analysis provided by classic approaches is only local and is not adequate to really identify which feature trajectories are relevant for the camera movement estimation.

To prove this point, we propose to run two preliminary experiments. We consider two videos :

- *close_person* : In this sequence, the person in the foreground is moving from side to side while being filmed by an unstable camera : see Figure 4.2
- 14_*person* : In this sequence, the train is moving forward rapidly while filmed by an unstable camera : see Figure 4.3

For these two videos, we run the RANSAC algorithm and watch for three successive frames the feature points labelled as inliers and outliers.

Figure 4.2 Three successive frames from the *close_person* sequence. Feature points classified as inliers by RANSAC are marked in green, while feature points classified as outliers are marked in blue. In the area circled in red, we can see that most feature points located on the person or in upper left corner change status abruptly.

- For the video *close_person*, we see on Figure 4.2 that in three adjacent frames, the RANSAC algorithm successively fits the motion model to different areas. In the first frame, it works perfectly as the moving person is correctly identified as an outlier and the motion fits over the entire background. In the second, the person is no longer identified as an outlier, and the motion model instead finds a compromise between the moving person and the right-hand portion of the background, which does not fit with the left-hand portion of the background. This means that including the person influences the motion model enough to reject part of the background. Finally, in the last frame, a motion model is found that fits both the background and the person.
- For the video 14_*object*, we see on Figure 4.3 that the features detected on the train start as outliers and are later classified as inliers when their perceived motion has slowed down. The transition between inlier and outlier can occur for large groups of features at once, as shown by the transition between the first and second frame. This abrupt change impacts the background, as a large portion of the background rightfully considered inliers in

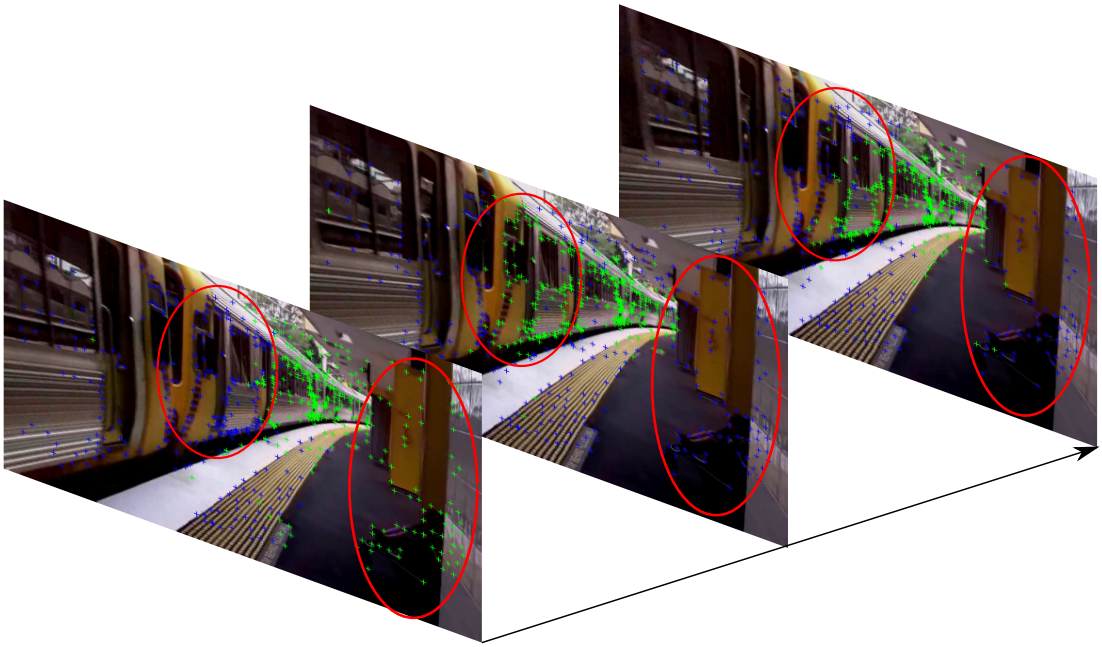

Figure 4.3 Three successive frames from the 14_*object* sequence. Feature points classified as inliers by RANSAC are marked in green, while feature points classified as outliers are marked in blue. Features identified on the train are initially classified as outliers while their perceived velocity is high and are progressively classified as inliers as their velocity in the camera plane decreases. The left hand circle highlights the area were most of thses features change from outliers to inliers. The right hand circle highlights features belonging to immobile objects, that change classification depending on the best compromise that the RANSAC algorithm can find between the background and the train.

the first frame are suddenly considered outliers. Since RANSAC cannot account for the motion between more than two frames, the incoherence of the movements of the train are ignored, and the best model found is a compromise between the motion of the train and the background - a compromise that can change very quickly from frame to frame.

This phenomenon is not only visible on several adjacent frames and is in fact very common when the RANSAC algorithm is used. To prove this point, we have computed for each frame t, the percentage of features points that were considered as inliers in frame t and outliers in frame $t+1$ (and vice versa). These plots are displayed in Figure 4.4 (for *close_person*) and Figure 4.5 (for 14_*object*). For the sequence *close_person*, we can see that we have frequent peaks where up to 65% of the features change from inliers to outliers and vice-versa, corresponding to frames where a compromise between the foreground and background is preferred to the background. This can mislead the camera motion

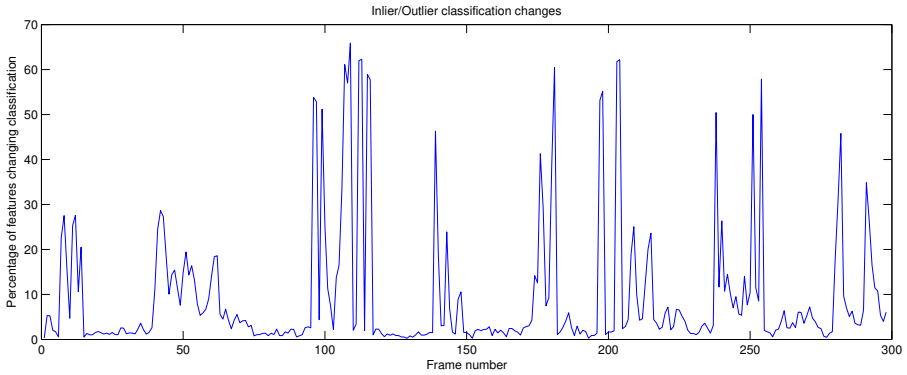

Figure 4.4 Percentage of features for which the classification inlier/outlier changes in the *close_person* video

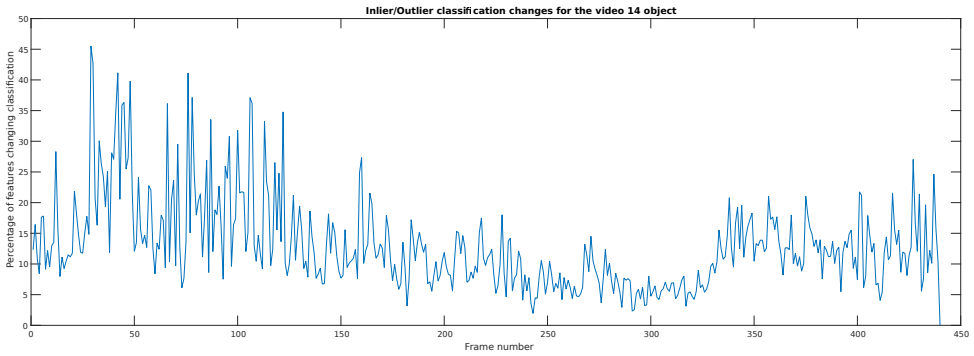

Figure 4.5 Percentage of features for which the classification inlier/outlier changes in the 14_*object* video

estimation into detecting sudden shifts in the camera movements as different parts of the scene are considered. In the sequence 14_*object*, fewer features change at one time but the changes are far more frequent, as the RANSAC algorithm constantly find a different balance between foreground and background. These instabilities highlight the benefits of fixing once and for all whether a given trajectory should be considered for the motion estimation rather than determining the inliers for every pair of frames.

4.1.3 Contributions

In this chapter, we propose a novel approach to assess and select the best feature trajectories to use in the camera motion estimation for video stabilization. Unlike standard approaches used for the selection of feature trajectories, we analyze the movement of the feature trajectories through all frames and compute a global

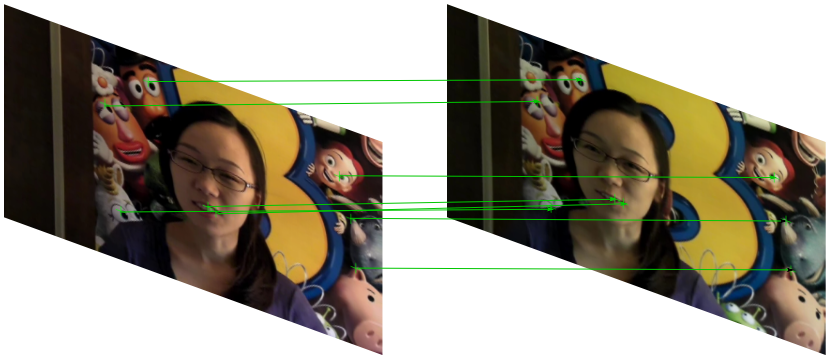

Figure 4.6 Illustration of the KLT feature points on the *close_person* video

weight by considering multiple criteria such as movement and duration. Section 4.2 presents the proposed method for feature trajectories selection. Section 4.3 is devoted to the performance evaluation, along with results on several videos and a comparison with state-of-the-art. The last section provide some concluding remarks and some open problems and perspectives.

4.2 Feature Trajectories Selection

First, let us consider a video corrupted by camera movements, from which feature trajectories are extracted using the KLT tracker [104]. This tracker detects interest points and tracks them throughout the video to form feature trajectories: this process is illustrated on Figure 4.6.

Let

$$\mathbf{z}_i[t] = (x_i[t],\ y_i[t])^\dagger \tag{4.1}$$

denote the position of the i^{th} feature point at frame t. The instantaneous velocity of this feature point is denoted

$$\dot{\mathbf{z}}_i[t] = \mathbf{z}_i[t+1] - \mathbf{z}_i[t]. \tag{4.2}$$

Since trajectory i might not last for the whole video duration, let define t_i^{start} and t_i^{end}, as the starting time and end time, respectively, of the i^{th} trajectory.

The feature trajectories selection strategy proposed in this chapter is based on the following steps:

1. First, we study each feature trajectory on a local time-window, in order to account for its duration and movement properties. More specifically, we

define two local weights $w_i^d[t]$ and $w_i^m[t]$ withing the range $[0, 1]$ that rank trajectory i according to its duration and its adequacy with the movements observed on a time-window centered on frame t.

2. Then, we merge all local weights $w_i^d[t], w_i^m[t]$ in order to form a global trajectory weight w_i that accounts for the phenomenon observed during the whole duration of the trajectory.

3. We select the feature trajectories with the largest weights w_i for the camera motion estimation.

4.2.1 Temporal criterion

Feature trajectories that span too few frames are likely to be unreliable. In most cases, they correspond to feature points that are not salient enough or not detected by the KLT tracker, or to moving objects that do not stay in the scene for long. This is a well-known problem usually handled by using *ad hoc* techniques such as duration thresholding in order to remove short trajectories and keeping only the longest ones[109].

To this end we propose to consider a time window of length $2N_w+1 = 31$ centered on frame t, and to compute a duration weight $w_i^d[t]$ that accounts for the local duration of trajectory i. This weight is defined as:

$$w_i^d[t] = \frac{\min(t_i^{end}, t + N_w) - \max(t_i^{start}, t - N_w) + 1}{2N_w + 1}. \tag{4.3}$$

This weight is within the range $[0, 1]$ and corresponds of the percentage of frames within the temporal window of interest for which trajectory i is defined.

4.2.2 Motion criterion

The aim of the second criterion is to associate to each feature point a weight related to the nature of its motion. The intent of this process is to discriminate between moving and static objects in the distorted video. However, without knowledge of the scene, the best motion model corresponding to the video is uncertain. Therefore, instead of using RANSAC [105] or its variants [53], which are based on parametric and geometric models, we propose to identify the dominant movement in the video without assuming a given motion model, by using a projection in a low-rank subspace[109].

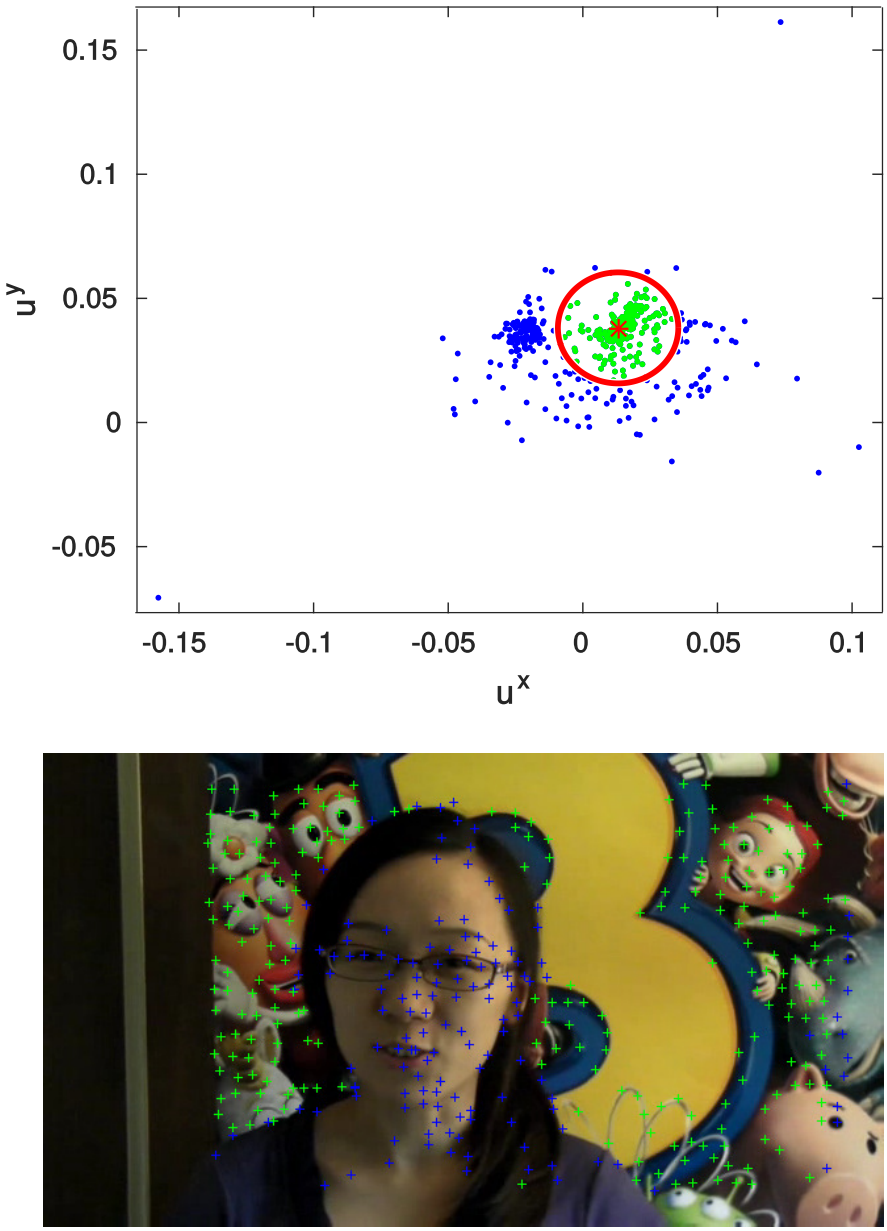

Figure 4.7 Example on the *close_person* video (frame 100). All trajectories belonging to the time-window of interest are projected in the $(\mathbf{u}^x,\mathbf{u}^y)$ plane. The majority of the contributions aggregate around the red point (mode of the 2D histogram). When selecting only feature points close to this mode (green points), we retrieve feature points from the background that are only corrupted by camera motion. On the contrary feature points far from the mode (blue points) correspond either to the moving woman or to spurious feature points.

We consider a time window of length $2N_w + 1 = 31$ centered on frame t, and form the local velocity matrix $\dot{\mathbf{Z}}[t]$ which contains all instantaneous velocities $\{\dot{\mathbf{z}}_i[\tau]\}_{\tau=t-N_w}^{t+N_w}$ belonging to trajectories that overlap with the time-window of interest. This matrix is then analyzed with a Singular Value Decomposition (SVD) algorithm that handles missing values through an iterative process [110] :

$$\dot{\mathbf{Z}}[t] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\dagger} \tag{4.4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and $\mathbf{\Sigma}$ is a diagonal matrix.

The largest singular value λ_1 in $\mathbf{\Sigma}$ captures the information corresponding to the dominant movement that is localized within the time-window of interest. The first left-singular vector $\mathbf{u}_1$ in $\mathbf{U}$ corresponds to the contribution of each trajectory to this dominant movement. Intuitively, all feature trajectories belonging to static objects or background, should have similar contributions to this dominant movement. On the contrary, moving objects or pixels should have different contributions and thus be identifiable. First, $\mathbf{u}_1$ is decomposed by taking every other line and forming $\mathbf{u}^x$ and $\mathbf{u}^y$, with $\mathbf{u}^x$ containing the parameters describing the parameters of the horizontal movements and $\mathbf{u}^y$ the parameters of the vertical movements. Figure 4.7 illustrates this process: feature points affected only by camera motion tend to aggregate in the $(\mathbf{u}^y, \mathbf{u}^x)$ plane. By detecting the mode (u_*^x, u_*^y) of the 2D-histogram (16×16) of these contributions, we can define a movement weight $w_i^m[t]$ that accounts for the distance (in the $(\mathbf{u}^y, \mathbf{u}^x)$ plane) of trajectory i to the detected mode:

$$w_i^m[t] = e^{-\gamma\left[(u_i^x - u_*^x)^2 + (u_i^y - u_*^y)^2\right]} \tag{4.5}$$

where $\gamma = 1000$ is a scale parameter. This weight is comprised between 0 and 1 and can be interpreted as an agreement score according to the dominant movement. It is a local score that provides a non-parametric assessment of the adequacy of the movement of the feature point.

4.2.3 Combination process

Both local weights $w_i^d[t]$ and $w_i^m[t]$ provide complementary information on the relevance of the feature trajectories. For instance, long trajectories may correspond to moving objects and have large duration weights, but are likely to have small movement weights since their movements would not fit the dominant motions seen in the video. It is therefore intuitive to combine both scores in order to take into account both criteria in the selection process.

Although local weights can provide insights on the relevance of the trajectories, they are not sufficient to select the feature trajectories. For example, a moving object can be static or follow the camera motion through a few frames and then return to its original own movement. In this case, the local movement weight increases and then decreases in the video, despite the fact that the trajectory is not suitable for robust camera parameter estimation. Note that all common methods for feature trajectory selection (such as RANSAC) have the same drawbacks. Indeed, since they consider only two successive frames, they might consider as inliers feature points that belong to moving objects but are static in the few frames of interest (see Section 4.1.2).

In order to address these two issues, we propose to first combine the two local weights, and then average the obtained local weight on the whole duration of the trajectory. This leads to the local weight defined below.

$$w_i = \frac{1}{t_i^{end} - t_i^{start} + 1} \sum_{t=t_i^{start}}^{t_i^{end}} w_i^d[t] \times w_i^m[t]. \tag{4.6}$$

This weight lies in the range $[0, 1]$. It is worth to point out that trajectories with large weights w_i have sufficient duration and their movements are consistent in accordance to the dominant movements present in the video. This means that it is unlikely that these trajectories belong to non detected KLT feature points, moving objects or artifacts.

This weighting method can be used in several strategies for feature trajectories selection. In this work, we remove all trajectories whose weight w_i is lower than a threshold λ. This threshold is set such as there is always a minimum number $N_s = 40$ of trajectories present in each frame.

4.3 Results and discussion

The selection process introduced in this chapter can be seen as a pre-processing step that can be used in any video stabilization method that relies on feature trajectories. This step can be evaluated independently by visual inspection of the selected trajectories or as part of a video stabilization process.

4.3.1 First observations

Figure 4.8 presents the selected trajectories for the video *close_person*. In this video, the woman is moving in front of a static background that is only corrupted by camera movements. The selection process successfully extracts $N_s = 40$ trajectories belonging to the static background and all those belonging to the moving foreground are removed. Note that, contrary to RANSAC, this selection remains the same during the whole video, so there is no risks of oscillations between inlier/outlier status.

Additional results on five different videos can be found on our webpage[1]: in all tested videos, the trajectories belonging to moving objects are correctly rejected by our method.

4.3.2 Evaluation framework

In the following, we investigate the relevance and the impact of the selection step within the video stabilization process. To that end, we propose to plug our preprocessing step into a standard video stabilization method, called Local Linear Matrix-Based smoothing (LLMB) [82]. First, we estimate the 2D affine model $H_{t,t+1}$ between two successive frames t and $t+1$

$$H_{t,t+1} = \begin{pmatrix} 1+a_{11} & a_{12} & T_x \\ a_{21} & 1+a_{22} & T_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.7}$$

These transformations are computed by solving a least-square problem from the feature trajectories retained by our trajectory selection algorithm. Then, these transformations are accumulated and smoothed using a Gaussian filter (with $\sigma = 50$). The difference between the accumulated and smoothed transformations defines an inverse transform $\tilde{H}_t$ that can be applied to frame t in order to diminish the camera movements. The resulting stabilized video is finally cropped to remove undefined regions.

For sake of comparison, three different stabilization methods are evaluated :

- In our method, 2D transforms are computed with a least-square approach by using only the selected trajectories. The video is then processed with the LLMB stabilization method [82].

[1] http://www-l2ti.univ-paris13.fr/~guilly/

Figure 4.8 Example on the *close_person* video (frame 100). On the left frame are all trajectories detected by the KLT tracker, and on the right are the selected trajectories. All trajectories corresponding to the moving woman have been removed.

- In the RANSAC method, 2D transforms are computed from all available trajectories using the RANSAC approach [105]. The video is then processed with the same LLMB stabilization method [82].
- In Youtube stabilizer, local outliers are removed and the video is stabilized by using cinematographic criteria [4].

4.3.3 Subjective evaluation

We tested these methods on fifteen videos, which are illustrated in figure 4.9 presenting different challenges for video stabilization, such as large moving object or depth differences [109]. An excerpt of these results is available on our webpage[1]. Figure 4.10 presents a screen-shot of the results obtained by the three methods for the *14 _ object* video which depicts a train leaving a station. Interestingly, the motion of the train is interpreted by both the RANSAC algorithm and the Youtube stabilizer as being part of the camera movement. The RANSAC algorithm seeks a compromise between the parameters of the background movement and the train motion, causing distortions in the video, that are outlined in green in the figure. The Youtube stabilizer avoids spatial distortions but causes artificial camera motion in an attempt to stabilize the train motion in short bursts before re-centering on the original camera path. These temporal distortions are visible on the video but are unfortunately impossible to display on still image frames. We recommend the readers to refer to our website for further analysis. Note that in both cases the attempt to correct the movements of an object causes the RANSAC and Youtube algorithm to stray further than necessary from the original camera path, causing a loss in resolution. This loss is easily visible for RANSAC and our method since cropped areas are visible in black, but it can also be seen in the Youtube result by comparing how much of the train is visible

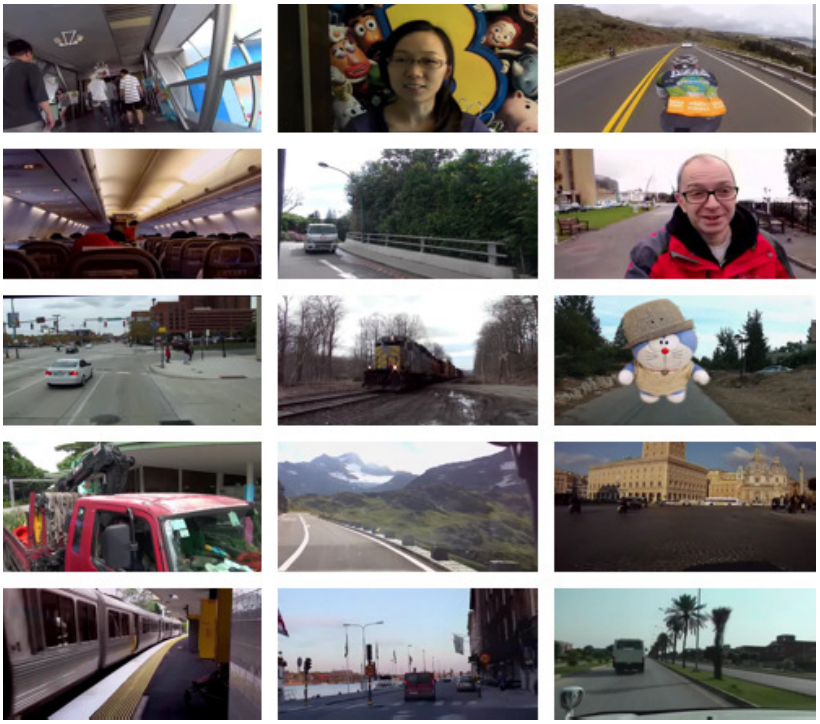

Figure 4.9 Sample thumbnails of the videos used to evaluate the proposed method.

compared to the original video. Results on other videos show similar effects: our method allows robustness in the presence of large moving objects for which both the RANSAC algorithm and the Youtube video editor fail.

4.3.4 Objective evaluation

In order to confirm the observed results, we need objective metrics to measure the benefits of the proposed selection method. We propose to use the objective metrics already described in Chapter 3 to compare the results of our feature selection compared to the results using the RANSAC algorithm.

Seven metrics are used for eveluation : Interframe Transformation Fidelity (ITF), Average Speed (AvSpeed), Average Acceleration (AvAcc), Average Percentage of

Figure 4.10 Example on the *14_object* video (frame 68). On the upper left is the original video, on the upper right the result of the RANSAC algorithm, on the lower left the results of the Youtube stabilizer and on the lower right the result of our algorithm. Cropped areas are displayed in black for RANSAC and our method. Youtube stabilizer automatically re-sizes the video.

Conserved Pixels (AvPCP) and Inter-frame Similarity Index (ISI), Spatial Efficient Entropic Differencing for Quality Assessment (SPEED-QA) [95] and VIIDEO [96]. For more explanations, please refer to Section 3.3.3. Results are displayed in Table 4.1.

Results shows improvements over the ITF metric for stabilized videos over the original video in most cases, with the exception of three videos containing particularly important depth differences that affine transforms cannot account for. Our selection method shows slight improvements on average over the standard method, with particular improvements over the video “9 object", where the RANSAC algorithm is mislead by the large moving object it contains whereas we correctly discard it. Two videos show better results for untreated videos: "10 driving" and "8 driving", both of which contain strong depth changes which are a poor fit for the affine model used. Similarly, the ISI metric favours the stabilized videos over the original, with the exception of videos containing strong depth differences. No significant differences can be found on average between our selection method and the RANSAC algorithm. The average speed AvSpeed shows on average better performances on our method but is highly variable from one video to the next. The average acceleration AvAcc usually favours the original video. One

Video	Version	ITF	ISI	AvSpeed	AvAcc	AvPCP	SPEED-QA	VIIDEO
3 crowd	*Original video*	18.27	0.61	3.59	**2.22**	**1**	0.86	**0.78**
	RANSAC	**19.64**	**0.66**	**2.14**	2.73	0.44	**0.077**	0.81
	Trajectory selection	19.58	**0.66**	2.15	2.67	0.43	0.082	0.81
4 object	*Original video*	26.45	0.85	2.22	**1.29**	**1**	0.090	**0.69**
	RANSAC	**28.89**	**0.90**	**1.31**	1.40	0.76	**0.037**	0.70
	Trajectory selection	28.85	**0.90**	**1.31**	1.39	0.77	0.039	0.70
22 driving	*Original video*	29.44	**0.94**	**0.39**	**0.60**	**1**	**0.023**	**0.66**
	RANSAC	**30.18**	**0.94**	0.46	0.76	0.87	**0.023**	0.68
	Trajectory selection	29.69	**0.94**	0.5	0.87	0.95	0.028	0.69
8 object	*Original video*	19.71	0.65	2.19	**0.64**	**1**	0.166	0.63
	RANSAC	22.64	0.72	**1.58**	0.86	0.70	0.071	**0.62**
	Trajectory selection	**22.87**	**0.75**	1.89	0.71	0.75	**0.061**	0.63
1 object	*Original video*	21.60	0.72	4.16	**2.36**	**1**	0.136	**0.76**
	RANSAC	**23.95**	**0.78**	**2.41**	2.74	0.61	0.098	0.77
	Trajectory selection	23.76	**0.78**	2.46	2.76	0.58	**0.086**	0.78
12 object	*Original video*	26.58	**0.86**	**0.77**	**0.43**	**1**	**0.085**	**0.59**
	RANSAC	**27.08**	**0.86**	0.89	0.68	0.96	0.095	0.63
	Trajectory selection	26.83	**0.86**	0.91	0.70	0.98	0.105	0.63
5 driving	*Original video*	**24.39**	**0.80**	**2.69**	**3.41**	**1**	**0.128**	**0.70**
	RANSAC	23.92	0.78	3.44	5.38	0.77	0.134	0.72
	Trajectory selection	23.84	0.78	3.47	5.48	0.79	0.141	0.72
17 driving	*Original video*	21.81	0.77	3.10	**1.60**	**1**	0.158	**0.72**
	RANSAC	22.43	0.80	2.31	2.13	0.62	0.043	**0.72**
	Trajectory selection	**22.77**	**0.81**	**2.23**	2.04	0.63	0.046	**0.72**
8 driving	*Original video*	**31.71**	**0.84**	**0.55**	**1.65**	**1**	**0.009**	0.77
	RANSAC	24.38	0.78	1.04	2.50	0.79	0.034	**0.72**
	Trajectory selection	24.48	0.78	0.96	2.36	0.82	0.034	0.75
9 object	*Original video*	17.40	0.48	3.43	**1.35**	**1**	**0.159**	**0.57**
	RANSAC	17.19	0.51	3.87	4.44	0.57	0.187	0.68
	Trajectory selection	**18.78**	**0.57**	**2.41**	1.73	0.79	0.163	0.60
20 driving	*Original video*	23.34	0.76	2.11	**1.16**	**1**	**0.098**	**0.70**
	RANSAC	**23.95**	**0.77**	**2.03**	1.75	0.77	0.099	0.73
	Trajectory selection	23.84	0.77	2.08	1.90	0.82	0.104	0.72
10 driving	*Original video*	**24.63**	**0.83**	**2.18**	**2.43**	**1**	**0.121**	**0.70**
	RANSAC	21.61	0.81	2.48	3.37	0.70	0.135	0.74
	Trajectory selection	21.73	0.82	2.44	3.47	0.77	0.139	0.74
14 object	*Original video*	20.72	0.71	1.04	**0.78**	**1**	**0.066**	**0.54**
	RANSAC	21.23	0.71	**1.02**	1.02	0.64	0.069	0.66
	Trajectory selection	**21.46**	**0.73**	**1.02**	0.95	0.89	**0.066**	0.60
10 object	*Original video*	20.48	0.62	2.66	0.87	**1**	0.065	0.64
	RANSAC	**23.44**	**0.73**	**1.79**	0.70	0.82	**0.063**	**0.62**
	Trajectory selection	23.31	0.72	1.89	**0.68**	0.82	0.069	0.63
15 object	*Original video*	25.1	**0.76**	2.11	**2.22**	**1**	0.109	0.81
	RANSAC	**22.47**	**0.76**	**1.89**	3.06	0.77	0.090	**0.76**
	Trajectory selection	22.32	0.72	**1.89**	2.98	0.77	**0.089**	0.77
Average	*Original video*	23.44	0.75	2.18	**1.53**	**1**	0.100	**0.68**
	RANSAC	23.53	**0.77**	1.91	2.24	0.72	**0.083**	0.70
	Trajectory selection	**23.61**	**0.77**	**1.84**	2.04	0.77	0.085	0.70

Table 4.1 Full results on the database. The best results are shown in bold.

possible explanation is that the stabilization process seeks to diminish the amplitude of the camera movements rather than its variations. Our method, while significantly worse than the original videos, does provide an improvement over the method using RANSAC. The cropping ratio AvPCP is significantly better using our algorithm, obtaining on average 5% more resolution than the alternative, with improvements up to 25% on some of the more challenging videos. The original video is of course not cropped, and so retains the full resolution. SPEED-QA results are better for the stabilized videos on average, despite several original videos being rated above the stabilized versions. The method using RANSAC slightly outperforms our approach on average. The VIIDEO metric rates the original videos better, both on average and in most cases. This could be linked to distortions caused by unsuitable transformations. In most cases the differences our approach and the standard stabilization methods are extremely small, and no differences can be observed on average. Both video and image quality metrics show very close results between RANSAC-based stabilization and our method, which can be explained by the fact that the stabilization methods are very similar. In particular, the use of an affine model can strongly impact the image structure that is evaluated by those metrics. improvements over the RANSAC-based method are more pronounced on the metrics AvAcc, AvSpeed and AvPCP, although interestingly the AvAcc is better for the original videos.

Some authors [82] have recently introduced an unsupervised and objective criterion for the evaluation of video stabilization, which is based on resolution loss. The idea is to compare the percentage of empty regions obtained by several methods with the exact same level of stabilization (σ value). Indeed, different degrees of stabilization naturally lead to differences in resolution loss. However by comparing this percentage on videos treated with the same stabilization method, we can judge whether our trajectory selection helps limit the resolution loss. Intuitively, this criterion illustrates how close the transformations estimated using our trajectory selection are to the true camera movement compared to the transformations computed using RANSAC.

This metric is linked to the Average Percentage of Conserved Pixels (AvPCP), but not identical. It should be noted that the cropping strategy, which seeks a rectangular area of fixed dimensions, lead to different results between the percentage of undefined area and cropped ratio. Enforcing a rectangular shape often leads to crop some valid areas of a frame to ensure that we retain a rectangle, while fixing the dimensions of the window leads to determining the cropping ratio when using the maximum rather than the average of undefined areas. Results for this metric on fifteen videos are presented on Table 4.2 for our method and the RANSAC method. Unfortunately, results are not available for Youtube stabilizer as it uses different stabilization and cropping strategies. Table 4.2 shows

Video	RANSAC	Trajectory selection
3_crowd	8.69	**8.09**
4_object	4.17	**3.89**
22_driving	3.72	**1.15**
8_object	10.75	**3.83**
14_object	3.90	**1.43**
close_person	**6.55**	6.57
10_object	**3.72**	3.77
12_object	0.80	**0.43**
5_driving	**3.14**	3.33
17_driving	9.74	**6.86**
8_driving	2.82	**2.36**
15_object	**4.34**	4.65
9_object	7.42	**3.32**
20_driving	2.99	**2.79**
10_driving	4.53	**2.62**
average	5.15	**3.67**

Table 4.2 Mean percentage of undefined area before cropping

that using our trajectory selection reduces the average undefined area (-1.5% in average), which leads to better resolution after cropping. In particular, scenes containing large moving objects greatly benefit from our method, making it possible to obtain a strong stabilization while keeping acceptable video resolution. The video "8_object" is the clearest example. In this video, a truck in the foreground moves forward and right, exiting the scene before the end of the video. At the time the truck leaves the scene, the stabilization method using RANSAC creates a sudden distortion, with an extremely strong shearing effect that results in large undefined areas. This is due to the transition between a motion model that found a compromise between the motion of the truck and that of features in the background to a motion model based solely on feature in the background. On the other hand, our method avoids using features located on the truck from the beginning of the sequence, so no such distortion is observed, removing the undefined areas it caused. This leads to an improvement of 3.83% of undefined pixel using our selection method compared to 10.75% using RANSAC. Similar videos are "22_driving" and "10_driving", both featuring large objects appearing or disappearing from view. In "9_object" the RANSAC algorithm mistakes the movements of a large object as the dominant motion, whereas our method correctly identifies the background. In several cases with large objects, such as "15_object", RANSAC obtains similar or better results, as it also identifies the moving objects but avoids over-fitting to a specific area of the scene.

4.4 Conclusion

In this work, we presented a feature trajectories selection method for video stabilization. Through this study, it has been shown that by taking into account duration and motion criteria, it was possible to select more reliable feature trajectories to be used for video stabilization purpose. The obtained results have been evaluated subjectively and objectively using some intuitive criteria. The proposed method shows smaller percentage of undefined areas using similar stabilization methods. Future perspectives include making use of the spatio-temporal distribution of feature points to refine the selection process and the impact of different motion models on the performance.

Chapter 5
Conclusion

Chapter 5

Conclusion

Through this thesis, it has been shown that although there are many published theoretical works and efficient software and hardware solutions for video stabilization, there are still many challenging issues that need to be tackled. From the results and the provided comprehensive overview, we could conclude that VS is far from being a solved problem. One typical example, is to consider a video taken from a moving car, facing the front of the vehicle. Such scenes contain large depth differences that make the computation of the camera motion difficult, as 2 dimensional models will not be able to account for large depth differences. While 3 dimensional models can account for such depth changes, they are very susceptible to outliers, such as moving objects like other vehicles on the road. Another example is a shot tracking a group from a close distance. If this group takes up a large enough portion of the frame, it is easy to mistakenly use their motion to determine the motion of the camera. Indeed, outlier detection methods often rely on outliers having different movements, and struggle in the presence of an homogeneous group of moving objects or people. This can result in detecting erroneous camera movements and worsen the camera motion as we compensate the wrong motions. This thesis has also clearly pointed out the lack of an effective framework for VSQA. This lack of standardized evaluation impairs both the evaluation of stabilization method and investigations in what types of camera motion are desirable, as well as when the stabilization artifacts are acceptable and when they are worse than the original unintentional motions.

This thesis attempts to contribute to the lack of standardized methods of VSQA by reviewing different objective quality assessment measures that have been proposed in the literature and assess their reliability when compared with a subjective evaluation study. This is done using four representative methods of video stabilization, but we also examine the results on the original video to better understand the extent to which video stabilization artifacts can be detrimental to users. Additionally, we propose a feature trajectories selection method for video stabilization, in order to deal with scene containing important moving objects.

This selection method makes no assumptions regarding the type of camera motion expected, making it compatible with any stabilization methods using feature tracking. The obtained results are evaluated subjectively and objectively using the previously investigated metrics. By using temporal criteria and an analysis over time of the motion of feature points, we can eliminate undesirable features corresponding to moving objects. Several metrics have shown that this allows improvements to the stabilization results.

Currently, perceptual models seem to be the most promising option to model the camera motion, as 2D models have proven too rigid for a number of scenes while 3D models remain very susceptible to outliers. In particular, a model using homography mixtures has been implemented in adobe after-effects and is among the state-of-the-art [76]. While optical flow has seen renewed interest in the light of these new types of model, the computational cost remains an obstacle, while feature tracking using SURF matching or the KLT tracker are very effective. However, the detection and removal of outliers remains a challenge for extreme cases where immobile features are outnumbered by moving features. In such cases, it is difficult to correctly differentiate between the moving objects and the fixed background. Furthermore, the evaluation of stabilization methods still lacks a reference metric that correlates well to viewer preferences. One way to go a step further and propose new solutions is to exploit the new advances in machine learning approaches and more specifically Deep Learning methods. Indeed, combining DL techniques with perceptual vision models and especially motion perception models is undoubtedly a promising approach to propose a effective evaluation method. Deep learning could also be exploited to recognize common moving objects without relying on the background forming the majority of the scene.

Bibliography

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

[2] M. L. Gleicher and F. Liu, "Re-cinematography: Improving the camera dynamics of casual video", in *Proceedings of the ACM International Conference on Multimedia*, 2007, pp. 27–36.

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

[4] M. Grundmann, V. Kwatra, and I. Essa, "Auto-directed video stabilization with robust L1 optimal camera paths", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 225–232.

[5] D. Sachs, S. Nasiri, and D. Goehl, "Image stabilization technology overview", *InvenSense Whitepaper*, 2006.

[6] Q. Zheng and M. Yang, "A video stabilization method based on inter-frame image matching score", *Global Journal of Computer Science and Technology*, 2017.

[7] C. Jia and B. 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)*, 2012, pp. 203–208.

[8] Y.-M. Liang, H.-R. Tyan, S.-L. Chang, H.-Y. Liao, and S.-W. Chen, "Video stabilization for a camcorder mounted on a moving vehicle", *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1636–1648, 2004.

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

[10] E. Ringaby and P.-E. Forssén, "Efficient video rectification and stabilisation for cell-phones", *International Journal of Computer Vision*, vol. 96, no. 3, pp. 335–352, 2012.

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

[12] S. Jeon, I. Yoon, J. Jang, S. Yang, J. Kim, and J. Paik, "Robust video stabilization using particle keypoint update and L1-optimized camera path", *Sensors*, vol. 17, no. 2, p. 337, 2017.

[13] J. Sánchez, "Comparison of motion smoothing strategies for video stabilization using parametric models", *Image Processing Online*, vol. 7, pp. 309–346, 2017.

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

[15] F. Liu, Y. Niu, and H. Jin, "Joint subspace stabilization for stereoscopic video", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 73–80.

[16] C. Morimoto and R. Chellappa, "Fast electronic digital image stabilization", in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, vol. 3, 1996, pp. 284–288.

[17] G. Zhang, X. Qin, W. Hua, T.-T. Wong, P.-A. Heng, and H. Bao, "Robust metric reconstruction from challenging video sequences", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[18] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization", *ACM Transactions on Graphics (TOG)*, vol. 30, no. 1, 4:1–4:10, 2011.

[19] N. Ahuja and R. Charan, "Pixel matching and motion segementation in image sequences", in *Recent Developments in Computer Vision, Second Asian Conference on Computer Vision, ACCV '95, Singapore, December 5-8, 1995, Invited Session Papers*, 1995, pp. 139–148.

[20] M. Irani and P. Anandan, "A unified approach to moving object detection in 2d and 3d scenes", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 577–589, 1998.

[21] H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 5, pp. 565–593, 1986.

[22] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques", *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[23] H. Farid and J. B. Woodward, "Video stabilization and enhancement", TR2007-605, Dartmouth College, Computer Science, Tech. Rep., 1997.

[24] A. Litvin, J. Konrad, and W. C. Karl, "Probabilistic video stabilization using kalman filtering and mosaicing", in *Image and Video Communications and Processing*, 2003, pp. 663–674.

[25] M. Wang, G.-Y. Yang, J.-K. Lin, S.-H. Zhang, A. Shamir, S.-P. Lu, and S. Hu, "Deep online video stabilization with multi-grid warping transformation learning", *IEEE Transactions on Image Processing*, vol. 28, pp. 2283–2292, 2018.

[26] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 81, 1981, pp. 674–679.

[27] B. K. Horn and B. G. Schunck, "Determining optical flow", *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[28] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them", *International Journal of Computer Vision*, vol. 106, no. 2, pp. 115–137, 2014.

[29] S. Liu, L. Yuan, P. Tan, and J. Sun, "Steadyflow: Spatially smooth optical flow for video stabilization", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 4209–4216.

[30] Y. Matsushita, E. Ofek, X. Tang, and H.-Y. Shum, "Full-frame video stabilization", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 50–57.

[31] Z. Wang and H. Huang, "Pixel-wise video stabilization", *Multimedia Tools and Applications*, vol. 75, no. 23, pp. 15 939–15 954, 2015.

[32] N. Tsoligkas, D Xu, I French, and Y Luo, "A motion model based video stabilisation algorithm", in *Proceedings of the World Automation Congress (WAC)*, 2006, pp. 1–6.

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

[34] A. Goldstein and R. Fattal, "Video stabilization using epipolar geometry", *ACM Transactions on Graphics (TOG)*, vol. 31, no. 5, p. 126, 2012.

[35] C. Tomasi and T. Kanade, *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[36] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, "SIFT features tracking for video stabilization", in *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, 2007, pp. 825–830.

[37] B.-Y. Chen, K.-Y. Lee, W.-T. Huang, and J.-S. Lin, "Capturing intention-based full-frame video stabilization", in *Computer Graphics Forum*, vol. 27, 2008, pp. 1805–1814.

[38] R. Hu, R. Shi, I.-f. Shen, and W. Chen, "Video stabilization using scale-invariant features", in *Proceedings of the International Conference on Information Visualization (IV)*, 2007, pp. 871–877.

[39] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung, "Video stabilization using robust feature trajectories", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 1397–1404.

[40] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)", *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[41] X. Zheng, C. Shaohui, W. Gang, and L. Jinlun, "Video stabilization system based on speeded-up robust features", in *Proceedings of the International Industrial Informatics and Computer Engineering Conference (IIICEC)*, 2015, pp. 1996–1998.

[42] C. Song, H. Zhao, W. Jing, and Y. Bi, "Robust video stabilization based on bounded path planning", in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, 2012, pp. 3684–3687.

[43] W. G. Aguilar and C. 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)*, 2014, pp. 1–6.

[44] S. Li, L. Yuan, J. Sun, and L. Quan, "Dual-feature warping-based motion model estimation", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4283–4291.

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

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

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

[48] Q. Ling, S. Deng, F. Li, Q. Huang, and X. Li, "A feedback-based robust video stabilization method for traffic videos", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 561–572, 2018.

[49] Q. Ling and M. Zhao, "Stabilization of traffic videos based on both foreground and background feature trajectories", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, pp. 1–1, Aug. 2018.

[50] C. Buehler, M. Bosse, and L. McMillan, "Non-metric image-based rendering for video stabilization", pp. 609–614, 2001.

[51] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[52] S. Choi, T. Kim, and W. Yu, "Robust video stabilization to outlier motion using adaptive ransac", in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 1897 –1902.

[53] L. Moisan, P. Moulon, and P. Monasse, "Automatic homographic registration of a pair of images, with a contrario elimination of outliers", *Image Processing On Line*, vol. 2, pp. 56–73, 2012.

[54] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng, "Meshflow: Minimum latency online video stabilization", vol. 9910, Oct. 2016, pp. 800–815.

[55] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 974–988, 2009.

[56] W. G. Aguilar and C. Angulo, "Real-time video stabilization without phantom movements for micro aerial vehicles", *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 46, 2014.

[57] S. Battiato, G. Puglisi, and A. Bruna, "A robust video stabilization system by adaptive motion vectors filtering", in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2008, pp. 373–376.

[58] H.-C. Chang, S.-H. Lai, and K.-R. Lu, "A robust and efficient video stabilization algorithm", in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 2004, pp. 29–32.

[59] X. Peng, J. Chen, and J. Zhang, "Robust digital image stabilization based on spatial-location-invariant criterion", in *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2011, pp. 2250–2254.

[60] G. Puglisi and S. Battiato, "Robust video stabilization approach based on a voting strategy", in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 629–632.

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

[62] J. Yang, D. Schonfeld, C. Chen, and M. Mohamed, "Online video stabilization based on particle filters", in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2006, pp. 1545–1548.

[63] M. Hansen, P. Anandan, K Dana, G Van der Wal, and P. Burt, "Real-time scene stabilization and mosaic construction", in *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 1994, pp. 54–62.

[64] P. Rawat and J. Singhai, "Adaptive motion smoothening for video stabilization", *International Journal of Computer Applications*, vol. 72, no. 20, 2013.

[65] D. Pang, H. Chen, and S. Halawa, "Efficient video stabilization with dual-tree complex wavelet transform", EE368 Project Report, Tech. Rep., 2010.

[66] M. Okade and P. K. Biswas, "Fast video stabilization in the compressed domain", in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2012, pp. 1015–1020.

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

[68] H. Qu and L. Song, "Video stabilization with L1-L2 optimization", in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 29–33.

[69] A. Nikolov and D. Dimov, "2D video stabilization for industrial high-speed cameras", *Cybernetics and Information Technologies*, vol. 15, no. 7, pp. 23–34, 2015.

[70] Z. Zhu, G. Xu, Y. Yang, and J. 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*, 1998, pp. 329–334.

[71] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3D video stabilization", *ACM Transactions on Graphics (ToG)*, vol. 28, no. 3, p. 44, 2009.

[72] Z. Zhou, H. Jin, and Y. Ma, "Plane-based content preserving warps for video stabilization", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2299–2306.

[73] C. Jia and B. L. Evans, "Constrained 3D rotation smoothing via global manifold regression for video stabilization", *IEEE Transactions on Signal Processing*, vol. 62, no. 13, pp. 3293–3304, 2014.

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

[75] D.-b. Lee, I.-h. Choi, B. C. Song, and T. H. Lee, "ROI-based video stabilization algorithm for hand-held cameras", in *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2012, pp. 314–318.

[76] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization", *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, 78:1–78:10, 2013.

[77] R. I. Hartley, "Euclidean reconstruction from uncalibrated views", in *Proceedings of the Joint European-US Workshop on Applications of Invariance in Computer Vision*, 1993, pp. 235–256.

[78] C. Tang and R. Wang, "Sparse moving factorization for subspace video stabilization", in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4314–4318.

[79] ——, "Local subspace video stabilization", in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2014, pp. 1–6.

[80] 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)*, 2017, pp. 1–6.

[81] G. Zhang, Z. Dong, J. Jia, L. Wan, T.-T. Wong, and H. Bao, "Refilming with depth-inferred videos", *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 828–840, 2009.

[82] J. Sánchez and J.-M. Morel, "Motion smoothing strategies for 2D video stabilization", *SIAM Journal on Imaging Sciences*, vol. 11, no. 1, pp. 219–251, 2018.

[83] N. 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*, vol. 86, 1998, pp. 120–131.

[84] S.-H. Yang and F.-M. Jheng, "An adaptive image stabilization technique", in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 3, 2006, pp. 1968–1973.

[85] G. Zhang, W. Hua, X. Qin, Y. Shao, and H. Bao, "Video stabilization based on a 3D perspective camera model", *The Visual Computer*, vol. 25, no. 11, pp. 997–1008, 2009.

[86] M. Niskanen, O. Silvén, and M. Tico, "Video stabilization performance assessment", in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2006, pp. 405–408.

[87] C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms", in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, 1998, pp. 2789–2792.

[88] L. Zhang, Q.-Z. Zheng, H.-K. Liu, and H. Huang, "Full-reference stability assessment of digital video stabilization based on riemannian metric", *IEEE Transactions on Image Processing*, vol. 27, pp. 1–1, Aug. 2018.

[89] T. T. Dang, A. Beghdadi, and M. Larabi, "A perceptual image completion approach based on a hierarchical optimization scheme", *Signal Processing*, vol. 103, pp. 127–141, 2014.

[90] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy, "Digital video stabilization and rolling shutter correction using gyroscopes", Stanford Tech Report CTSR 2011-03, Tech. Rep., 2011.

[91] L. Marcenaro, G. Vernazza, and C. S. Regazzoni, "Image stabilization algorithms for video-surveillance applications", in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2001, pp. 349–352.

[92] J. Xu, H.-w. Chang, S. Yang, and M. Wang, "Fast feature-based video stabilization without accumulative global motion estimation", *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, pp. 993–999, 2012.

[93] G. Thalin, "Deshaker", *Http://www.guthspot.se/video/deshaker.htm*, 2008.

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

[95] C. G. Bampis, P. Gupta, R. Soundararajan, and A. C. Bovik, "Speed-qa: Spatial efficient entropic differencing for image and video quality", *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1333–1337, 2017.

[96] A. Mittal, M. A. Saad, and A. C. Bovik, "A completely blind video integrity oracle", *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 289–300, 2016.

[97] Z. Wang, L. Lu, and A. Bovik, "Video quality assessment based on structural distortion measurement", *Signal Processing: Image Communication*, vol. 19, pp. 121–132, Feb. 2004.

[98] N. Barman, S. Schmidt, S. Zadtootaghaj, M. G. Martini, and S. Möller, "An evaluation of video quality assessment metrics for passive gaming video streaming", in *Proceedings of the 23rd Packet Video Workshop*, ser. PV '18, ACM, 2018, pp. 7–12.

[99] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "A subjective study to evaluate video quality assessment algorithms", in *Human Vision and Electronic Imaging*, 2010.

[100] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video", *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1427–1441, 2010.

[101] X. Min, K. Ma, K. Gu, G. Zhai, Z. Wang, and W. Lin, "Unified blind quality assessment of compressed natural, graphic, and screen content images", *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5462–5474, 2017.

[102] M. A. Qureshi, A. Beghdadi, and M. A. Deriche, "Towards the design of a consistent image contrast enhancement evaluation measure", *Signal Proceesing : Image Communication*, vol. 58, pp. 212–227, 2017.

[103] M. G. Kendall, "A new measure of rank correlation", *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[104] J. Shi and C. Tomasi, "Good features to track", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593–600.

[105] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[106] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization", *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, 78:1–78:10, 2013.

[107] ——, "Steadyflow: Spatially smooth optical flow for video stabilization", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 4209–4216.

[108] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung, "Video stabilization using robust feature trajectories", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 1397–1404.

[109] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization", *ACM Transactions on Graphics (TOG)*, vol. 30, no. 1, 4:1–4:10, 2011.

[110] N. Srebro and T. Jaakkola, "Weighted low-rank approximations", in *Proceedings of the International Conference on Machine Learning (ICML)*, 2003, pp. 720–727.