SIAM J. Imaging Sciences
Vol. 11, No. 1, pp. 219–251

Motion Smoothing Strategies for 2D Video Stabilization*

Javier Sánchez† and Jean-Michel Morel‡

Abstract. Video stabilization aims at removing the undesirable effects of camera motion by estimating its shake and applying a smoothing compensation. This paper proposes a unified mathematical analysis and classification of existing smoothing strategies. We assume that the apparent velocity induced by the camera is estimated as a set of global parametric models, typically those of a homography. We classify the existing smoothing strategies into compositional and additive methods and discuss their technical issues, particularly the definition of the boundary conditions. Our discussion of the various alternatives leads to clear-cut conclusions. It rules out the global compositional methods in favor of local linear methods and finds the adequate boundary conditions. We also show that the best smoothing strategy yields a scale-space analysis of the camera ego-motion parameters. Analyzing this scale-space on examples, we show how it is highly characteristic of the camera path, permitting us to compute ego-motion frequencies and to detect periodic ego-motions like walking or running.

Key words. video stabilization, motion compensation, motion smoothing

AMS subject classifications. 68T45, 15-04

DOI. 10.1137/17M1127156

1. Introduction.

Video stabilization is the process of compensating for the undesired motion produced by camera shake. Such a motion may be caused by several reasons, such as vibrations and harsh moves of human or vehicle borne cameras, hardware deficiencies in the camera components, looseness of the underlying platform, meteorological and environmental conditions, and uneven zooming. The goal of stabilization is to estimate the undesired motion and to warp the images to compensate for it. As we shall see, the stabilization signal is also a main characteristic of ego-motion, permitting us to analyze it without any extra calibration information on the camera path.

Stabilization is particularly useful for videos taken from hand-held cameras, where the camera jitter can be important. It is also interesting for film production and surveillance camera systems, where it may serve as an initial step for other high-level processes, such as background subtraction or object tracking.

It can be tackled from two different perspectives: *optical image stabilization* (OIS) and

*Received by the editors April 26, 2017; accepted for publication (in revised form) November 3, 2017; published electronically January 30, 2018.

http://www.siam.org/journals/siims/11-1/M112715.html

Funding: The work of the authors was partly funded by the BPIFrance and Région Ile de France, in the framework of the FUI 18 Plein Phare project, the European Research Council (advanced grant Twelve Labours), the Office of Naval research (ONR grant N00014-14-1-0023), and the Spanish Ministry of Economy, Industry and Competitiveness through the research project TIN2017-89881-R.

†Corresponding author. Centro de Tecnologías de la imagen (CTIM), Department of Computer Science, University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria 35017, Spain (jsanchez@ulpgc.es).

‡Centre de Mathématiques et Leurs Applications (CMLA), Ecole Normale Supérieure de Cachan, Université Paris-Saclay, Cachan 94235, France (morel@cmla.ens-cachan.fr).

electronic or digital image stabilization (EIS). OIS systems correct lens orientation or film sensor position in the camera to compensate for sudden movements by mechanical stabilizers. We deal here with EIS systems, which rely on the image sequence to estimate and compensate for the motion of the camera.

These techniques can be divided into three main steps: motion estimation, motion compensation or smoothing, and image warping. Additional postprocessing tasks are needed to deal with, for example, empty regions that appear at the border of the frames, or to improve the quality of the images after stabilization.

In the first step, the objective is to determine the camera motion along the scene. This can be difficult, or even impossible, so the objective is normally simplified by estimating the 2D transformations between the pixels of successive frames. These methods must be robust in the sense that they must deal with outliers, such as foreground moving objects or occlusions. The outcome of this step is a set of transformations that approximate the effect of the camera motion on the movie.

The aim of the motion compensation step is to use the transformations computed in the previous step to remove the undesired motion. The output of this process is a new set of transformations which, when applied to the original images, produce a more stable sequence. In this work, we focus on parametric motion models, which are the basis for the principal methods today. Our goal is to study and classify the various alternatives for motion smoothing. Our proposed classification will distinguish several video smoothing approaches, which can be organized into two main groups: *compositional* and *additive* strategies.

The former group relies on the composition of transformations and is, by far, the most widely used in the literature. They can be further divided into three different approaches: The *compositional approach* transforms all the images with respect to a fixed reference frame, for example the first one. It assumes that the camera is static and, therefore, eliminates any camera motion. It has been also used for building mosaics. The *global approach* introduces a simple smoothing process that preserves the main camera path. It removes the high frequency motion using a temporal convolution of the motion parameters with a Gaussian function. In the same way, it depends on an initial reference frame. Finally, the *local approach* does not rely on a unique reference frame. Each frame acts as its own reference system, so the regularization is carried out in a sliding time interval around the current temporal position.

In this article, we also propose two methods that belong to the *additive* group. In this case, the stabilization is not based on the composition of transformations but on a linear relation of their coefficients. These methods are based on the construction of incrementally computed *virtual trajectories*. The formulation of these strategies is equivalent to a linear scale-space theory.

Although the present paper is akin to a review format, it aims at selecting—and improving—the best found solution. We shall review (or detail) several almost equivalent smoothing strategies and fix the adequate boundary conditions. Furthermore, the quality of a video stabilization process also depends on the usage of the information that it provides on the video itself. We show that a good stabilization also delivers a multiscale eight parameter temporal stabilization signal, which we call a *stabilization scale-space*. This scale-space yields a reliable estimation of main intrinsic properties of the video, including a multiscale understanding of camera ego-motion parameters such as pitch, roll, yaw, and zoom, and of their main frequen-

cies in a sliding window. Requiring such scale-space properties leads us to select the adequate one among stabilization processes with similar properties.

Section 2 reviews the literature on the subject. Section 3 introduces the general video stabilization framework and fixes a common notation. Section 4 introduces the taxonomy of the motion compensation strategies, divided into the compositional and additive groups. Section 5 applies a simple criterion to select the best compensation strategy by measuring the empty space left in the video frame after compensation. Section 6 deduces from this choice a stabilization scale-space. Experiments demonstrate how it can be used to compute reliable ego-motion characteristics. Conclusions are presented in section 7.

2. A review of the relevant literature.

3D or 2D stabilization? Video stabilization techniques can be classified into 2D and 3D methods. The latter aim at finding the camera trajectory through external camera calibration and an estimation of the 3D structure of the scene. These are the most powerful strategies for stabilization because they can deal with the parallax motion [10] (but not necessarily with scenes with moving objects). The work in [31], for example, proposes a 3D method based on structure-from-motion (SFM) for creating stable hyperlapse videos.

The method of [59] performs a full 3D image reconstruction using a camera array with multiple viewpoints and estimates the camera path, which is thereafter smoothed to reduce the shaky effects. In short, multiview stereo is used to compute dense depth maps for each camera at each time instant. Using these depth maps, the output image sequence is generated from the desired virtual camera viewpoint. The method also requires knowledge of the camera calibration and the assumption that it does not change in the sequence. The authors acknowledge that, because a 3D reconstruction of the scene is a strong requirement, $2D$ *methods are still more preferable in practice*, which is our guideline in this paper.

The authors of [18], on the other hand, rely on the epipolar geometry and *avoid the perils of a* $3D$ *reconstruction*. Their method generates virtual point trajectories by using epipolar transfer functions and then applies a Gaussian smoothing to the virtual trajectories. The (sophisticated) frame warping is based on the correspondence between virtual and real trajectories. We shall review this stabilization technique (in the simpler case, though, where only correction homographies are considered).

The use of the information provided by depth cameras is another possibility, like in [44]. The recent work in [30] proposes the stabilization of 360º videos through the estimation of the relative 3D rotation between key frames, and then turns to a 2D optimization framework for the stabilization of the in-between frames.

As these papers indicate, estimating the 3D information is difficult because of several challenging tasks, such as the calibration of the cameras, fusing the information in the 3D scene from multiple views, or dealing with sparse data. Additionally, these techniques often rely on feature tracking, which may be challenging for unstable sequences.

For these reasons, several methods have tried to combine 2D and 3D strategies [40], but there is a plethora of methods based on feature tracking that we review in the next paragraph.

Trajectory smoothing methods. Researchers have indeed become increasingly aware of the risks of 3D stabilization techniques. For example, the authors of [41] notice that their

practicality is limited by the need to perform 3D reconstruction through SFM. They point out that these methods cannot work satisfactorily in many situations including lack of parallax, camera zooming, in-camera stabilization, and rolling shutter. Hence these same authors advocate a trajectory-based method. Features in the video are tracked and assembled into a "feature trajectory matrix," which is factored into two low-rank matrices: a coefficient matrix and an eigen-trajectories matrix. The method then performs a Gaussian smoothing of eigen-trajectories and obtains output trajectories by multiplying by the original coefficient matrix. This process ends with a video warping on the new feature trajectories.

Although our analysis here considers simpler methods, it will remain relevant for such feature trajectory tracking methods. Indeed, this sophisticated method again creates virtual trajectories and smooths them by a temporal moving Gaussian window to obtain a stabilization warping. As we shall see, our conclusions are similar to those of that paper, which recommends a sliding Gaussian convolution of virtual trajectories on about 50 frames. Our discussion of boundary conditions is directly useful to any practical application of this method.

In [34], it is argued that *homography-based schemes generally perform quite well for stabilizing planar scenes or rotational camera motions but suffer from highly nonplanar scenes.* In this case, the method does not require an estimation of camera motion. The authors argue, *our approach does not suffer from the problems with insufficient motion models and inaccurate motion classification.* It first detects a set of robust trajectories by feature tracking with spatial coherence, formulated as a first variational problem. These trajectories are the input used to estimate directly the parametric stabilization warping transforms, namely a temporal series of isometries or affinities. These are computed by minimizing a functional controlling both the roughness of the virtual trajectories and the degradation induced by warping. The roughness of the trajectories is controlled through their second derivative. Hence, it can be argued that the smoothing method is equivalent to a Laplacian smoothing and therefore to a Gaussian temporal filtering. Thus, our review here will be relevant for this method, particularly the discussion of the boundary conditions in the smoothing process.

The method in [67] represents feature trajectories using Bézier curves. It tracks trajectories across the video and regularizes them by minimizing the acceleration, while enforcing parallelism of similar trajectories. The warping is nonparametric and consists in the application of local homographies. Our study in this paper does not apply fully to this work, but still our discussion of boundary conditions to smooth trajectories is relevant, and the conclusions of this paper again point to the efficiency of Gaussian smoothing of virtual trajectories as the main stabilization tool.

2D strategies. Most current techniques are based on simpler 2D strategies that assume no external information about the 3D camera path. In other terms, only the effect of camera motion on the movie is handled—not the real 3D camera path. An estimate of the motion between successive frames is the only information used. Motion estimation techniques are divided into parametric motion models and optical flow methods. Dense optical flows provide much more information about the scene and may simultaneously track foreground objects and solid background. However, the stabilization cannot directly work with the true optical flow. It requires ad hoc manipulations to create a very smooth flow field. Moreover, current optical flow methods are slow and typically fail in the presence of homogeneous regions, large

displacements, or illumination changes. It follows that optical flow methods often end up being combined with parametric motion models to avoid creating nonrigid distortions [46].

Parametric models instead rely on global motions that are estimated as planar transformations with growing complexity, namely translations, similarities, affinities, or homographies. Homographies actually faithfully represent the effect on the image of a moving pinhole camera motion filming remote and steady objects, or the effect on the image of any rotation of the camera around its optical center. Applying a homographic stabilization therefore amounts to stabilizing either the whole image, or at least the background (remote) objects of the scene. For example, if the camera moves forward and the objects in front are remote, the background image deformation is a homothety.

Motion estimation. The techniques for parametric motion estimation can be classified into direct and feature-based methods. The former calculate the global motion by minimizing an energy functional, whereas the latter look for a set of salient points in the frames and compute the transformation that better puts these features in correspondence. The key idea is to find the displacement of the background of the scene, for which a homography is a good approximation, and discard potential foreground motions. Therefore, most techniques combine fast matching processes with robust strategies to discard outliers. Both direct and feature-based methods can be equipped with such robust strategies.

Classic methods [21, 52, 53] used correlation techniques and pyramidal structures for estimating large displacements. The correspondences were usually detected on a reduced number of blocks distributed on the images to improve the runtime. In order to remove outliers, simple rejection strategies were used, like cross-checking the correspondences. The method proposed in [66], for example, calculates block motion in two areas: the central region, which is associated with the foreground objects, and the background, which is considered to be near the border.

The combination of salient points, such as Harris corners [22], and the Kanade–Lucas–Tomasi (KLT) tracker [48, 58], has often been used. It is usually combined with a pyramidal structure for estimating large displacements [50, 20, 56]. Robust strategies like RANSAC are used for the rejection of outliers. In [42], the authors propose the use of motion vectors employed in the coding of video compression in order to accelerate the motion estimation step.

Affinities are arguably the most commonly used transformations for stabilization [21, 39, 50, 1], as they produce satisfying results without introducing deformations like with homographies. It is possible to use even more restrictive parameterizations such as translations [28, 66] or similarities [52]. Paradoxically, these often have a better performance in the presence of camera parallax, because they do not distort the images so much. Nevertheless, for more complex motions and arbitrary camera shakes, they cannot rectify the images as faithfully as affinities or homographies do. On the other hand, the 3D camera rotation model [15, 56] is also widely used. This is a restricted case of homography that works properly when the camera rotates about its optical center, and is still valid for small camera parallax.

Several works have proposed computing a set of transformations in each frame instead of a single parametric model. This is the case, for instance, of the *mixture of homography* model proposed in [19]. It relies on the KLT tracker and adapts a threshold to obtain many distributed features in the images. Various homographies are calculated for each frame in

different horizontal strips. This mixture is especially useful for dealing with the rolling shutter problem. Similarly, in [40, 45] the images are divided into cells of a regular grid. The algorithm computes a warping between the vertices of every two successive images, using the features and enforcing a similarity transform between the triangles. Finally, homographies are computed between corresponding cells.

Later, the same authors proposed a method for motion estimation based on smoothed optical flows [46], which is, however, computationally expensive. More recently, this was alleviated in [43] using a mesh of motion vector candidates instead of the full optical flow. These motion candidates are obtained through FAST features [65] and the KLT tracker and are processed by means of two median filters. In any case, these methods rely on parametric models for initializing the motion estimates.

In this work, our focus is not on the image matching method. In the experiments we used indifferently a classic feature-based technique [51], which relies on SIFT features [47], and a direct method [57], which implements the *inverse compositional algorithm* [5, 4]. Direct methods are usually faster but more sensitive to brightness changes and outliers. Feature-based methods depend on the type of selected features. They are typically more sensitive to noise and motion blur but allow us to estimate stronger deformations. A discussion of both strategies is given in [62].

There are many works for stabilization that rely on SIFT [47] or SURF [6] features, although these are slow for stabilization. In [54], the authors propose using MSER [49] features for estimating the motion between frames. The benefit of these features is that they are based on regions, which are typically more stable than features based on corners.

Motion compensation or smoothing. The objective of motion compensation is to estimate a set of transformations that, when applied to the original video, reduces the effect of camera shake. It should remove the camera jitter at the same time that it must preserve as much as possible the contents of the original sequence. Additionally, it should avoid introducing geometric distortions into the scene.

The simplest approach is to transform the images with respect to a reference frame. This is what we refer to as the *compositional approach* in section 4.1.1, and it was typical in the first methods, such as [52]. It is usually associated with the construction of mosaics and allows for real-time video stabilization [21]. This method is particularly valid if the camera is known to be static.

For arbitrary camera motions, the transformations must be smoothed. Techniques based on Gaussian smoothing are the most common [53, 50, 26]. These are more flexible than the previous approach, are easy to implement, and provide satisfactory results in general.

More recently, the introduction of optimization strategies, like the robust L^1 regularization strategy proposed in [20], has taken into consideration featured camera paths and the size of the cropping window. The method in [45], on the other hand, minimizes a functional that forces the new camera path to be close to the original path and contains a smoothness term to stabilize them. It also introduces a weight to preserve motion discontinuities due to fast motions of the camera. The technique in [11] approximates the smooth camera path by fitting the parameters of a polynomial curve. In [17] the camera trajectory is broken into different segments and a different smoothing strategy is applied in each segment depending on the type

of motion.

Image warping and postprocessing tasks. In the final step, the images are warped according to the calculated smoothed motion. *Crop & Zoom* is the most widely used strategy for eliminating empty regions. The objective is to find the maximum visible rectangle in the whole sequence and to apply a common transformation to the images. This transformation is usually the composition of a translation and scaling and can be carried out before the warping process. The result is a video with a shorter field of view and lower quality due to the zoom-in. For these reasons, it is important for the methods to maximize the crop region. The method in [19], for instance, allows the user to specify the size of the crop window and manages to find the smooth camera path that respects this size.

A different alternative, called *video completion*, consists in filling this information up from other frames. For example, the work in [39] uses mosaicing to fill up the missing image areas. However, this process is very complex in practice, because it depends on the motion of the camera and the structure of the scene. The result is that it typically introduces important artifacts at the border of the images. The method in [69] proposes filling the holes by sampling spatiotemporal patches in the same video. More sophisticated image inpainting techniques [12, 3] may be chosen, although the artifacts are more noticeable in videos.

The method in [50] proposes a filling strategy based on the optical flow, called *motion inpainting*. It calculates the optical flow between consecutive frames and propagates the information to the empty regions using the motion information. This is used to guide the warping of the image from neighbor frames in the case when a mosaicing does not provide reliable information.

Combining pixels from different frames can introduce brightness changes on the same image, so it is necessary to blend the information carefully, like in [63] or [9]. On the other hand, it is also interesting to correct global brightness changes produced by *varying exposures*, since this effect is unpleasant after stabilization.

Motion blur is also more noticeable after stabilization. In this case, image deblurring techniques are useful to improve the quality of the output video. Deblurring using deconvolution is in general difficult [32], and good solutions can be attained only if the camera shake is small. The method in [50] proposes an algorithm for improving the quality of the images by transferring sharp pixels between neighboring frames. In this case, the global motion is used both for image alignment and deblurring, whereas local motion is used for video completion.

The skew and wobble problems caused by the rolling shutter are important for video stabilization. These may introduce distortions on the images in the form of slanted objects or jelly effects. The rolling shutter compensation raises two issues: first, the kind of deformations are generated not only by the camera jitter, but also by the acquisition device, which are many times related; second, a homography model is not sufficient, and more general parametric correction must be considered. Many of the aforementioned works deal with this problem [24]. The work in [7], for example, deals with the rolling shutter problem using the information of on-board gyroscopes, which provide an estimate of the instant velocity of the camera.

In this work, we shall not deal with the rolling shutter problem and assume that the images are taken from a camera with a global shutter. Indeed, a correct formalism must first address the problem of global stabilization before extending it to roller shutter effects. Besides, these

are usually compensated locally by the same stabilization strategies.

Scale-space and the frequency analysis. Scale space theory started with the founding works of Witkin [70] and Koenderink [29]. They proposed analyzing a signal or an image by convolving it with Gaussians of growing standard deviations and noticed that this is equivalent to applying the heat equation. The notion of scale-space means that an event in the signal or image happens at a certain position, but also at a certain scale. The uniqueness of the heat equation satisfying a scale-space axiomatics is proven in [2, 37] and summarized in the review [68]. The main axioms leading to uniqueness are linearity, locality, translation invariance, rotation invariance (for images), and causality. Causality roughly means that no new detail is being created by the successive filters. Under these axioms the only possible scale-space process is the heat equation.

The theory of scale-space remains so fundamental that it is the object of several recent books [60, 38, 64]. After the founding works of Lindeberg [35, 36], scale-space established itself in image analysis as a way to detect local features, notably blobs detected by computing the 3D extrema of the normalized image Laplacian. The scale of these extrema was shown by Lindeberg to be proportional to the blob's size. This method had found a groundbreaking application with the SIFT method [47], by now used by all image analysis practitioners. The SIFT method was generalized to video and movies by Laptev and Lindeberg [33]. Scale-space has also been proposed to stabilize the video gray scale distribution and to compensate for flickering effects. In that case the heat equation is applied to an inverse of the frames' gray level cumulative distribution function [14].

Several works have used parametric models between two successive frames in the video for the purpose of video indexing, camera motion characterization, or ego-motion classification, such as [8] or [27]. The scale-space analysis that we propose for video stabilization can also be used for other related tasks, such as activity classification and detection [25, 61, 55].

3. The video stabilization formalism. The motion estimation step receives a sequence of images, $\{\mathbf{I}_i\}$, and computes a set of transformations, $\{\mathbf{H}_{i,i+1}\}$, between successive frames, i and $i+1$, as illustrated in Figure 1. The pixels of the images are related by the photometry consistency principle $\mathbf{I}_i(\mathbf{x}) = \mathbf{I}_{i+1}(\mathbf{H}_{i,i+1}\mathbf{x})$, where $\mathbf{x} = (x, y, 1)$ is the pixel position. These matrices may be, for example, any of the transformations given in Table 1, each one depending on a set of parameters. These parameterizations are typical of direct methods [62]. Note that the matrices and points are expressed in homogeneous coordinates, so the image positions are obtained after normalizing by the third component. In the following, we will be using indistinctly matrix notation or its corresponding parameterization.

The motion smoothing step obtains a new set of transformations, $\{\mathbf{H}'_i\}$, from the computed motions $\{\mathbf{H}_{i,i+1}\}$. To that purpose, it applies a Gaussian convolution to the time series of transformations. We discuss the smoothing alternatives in the following section. $\{\mathbf{H}'_i\}$ are the transformations that must be applied to $\{\mathbf{I}_i\}$ to eventually obtain the stabilized images $\{\mathbf{I}'_i\}$.

The postprocessing step is the process necessary to remove the empty regions that appear at the border of the frames after image warping. The last step consists in applying the smoothing transformations as

$$\mathbf{I}'_i(\mathbf{x}) := \mathbf{I}_i(\mathbf{H}'_i\mathbf{x}). \tag{3.1}$$

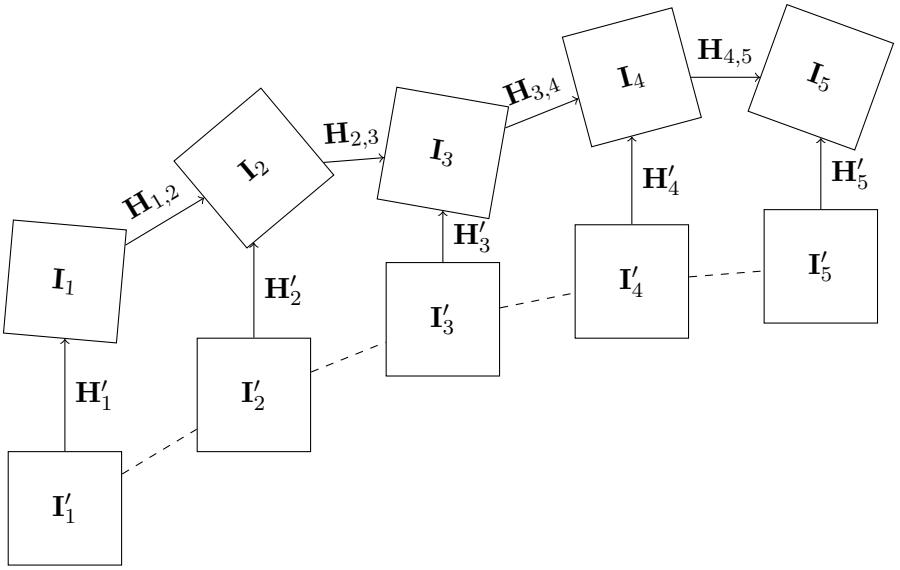

Figure 1. *Illustration of the motion smoothing process.* $\{\mathbf{I}_i\}_{i=1,\dots,N}$ *is the initial image sequence;* $\{\mathbf{H}_{i,i+1}\}_{i=1,\dots,N-1}$ *are the computed transformations between consecutive images.* $\{\mathbf{I}'_i\}_{i=1,\dots,N}$ *is the stabilized sequence, and* $\{\mathbf{H}'_i\}_{i=1,\dots,N}$ *are the stabilizing transformations.*

Table 1

Typical planar transformations with their parameters and homogeneous matrix representation.

Transform	**Parameters** ($\mathbf{p}$)	**H**
Translation	(t_x, t_y)	$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$
Euclidean	(t_x, t_y, θ)	$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix}$
Similarity	(t_x, t_y, a, b)	$\begin{pmatrix} 1+a & -b & t_x \\ b & 1+a & t_y \\ 0 & 0 & 1 \end{pmatrix}$
Affinity	$(t_x, t_y, a_{11}, a_{12}, a_{21}, a_{22})$	$\begin{pmatrix} 1+a_{11} & a_{12} & t_x \\ a_{21} & 1+a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}$
Homography	$(h_{11}, h_{12}, h_{13}, \cdots, h_{32})$	$\begin{pmatrix} 1+h_{11} & h_{12} & h_{13} \\ h_{21} & 1+h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}$

Figure 2 shows an example using Euclidean versus homographic transformations for video stabilization. The van passes very close to the camera, and, for a fraction of second, the foreground moving object dominates the scene. The feature-based homography detection mixes foreground and background and leads to a stabilization attempt focused on foreground. The estimate of Euclidean transformations also mixes foreground and background in such a situation. Yet the Euclidean transforms do not deform the scene so much. They are therefore often preferred to handle sequences with alternating foreground/background.

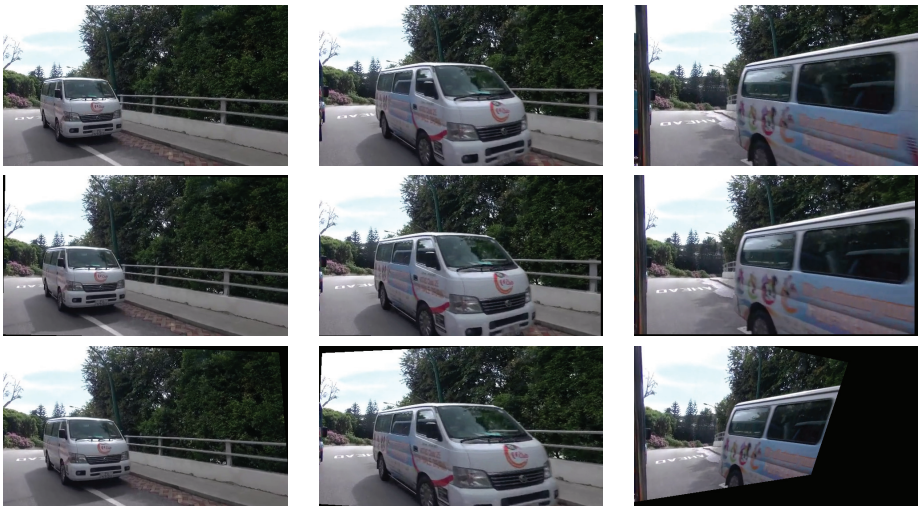

Figure 2. *Foreground motion: Comparison of Euclidean and homographic transformations for video stabilization. First row, original sequence (frames* 50, 75, 100*); second row, stabilization with Euclidean transforms; third row, using homographies. A direct method was used for estimating the motion and the local matrix–based smoothing scheme presented below for motion compensation. The homographies mix up foreground and background and attempt to compensate the van's deformation. The Euclidean compensation is more conservative.*

4. Motion compensation. The objective of motion compensation is to obtain a new set of transformations that effectively remove the undesired background motion. Given the transition background transforms $\{\mathbf{H}_{i,i+1}\}$, which we assume henceforth to be correct, the aim is to obtain the stabilization deformations, as in Figure 1. There are several strategies for computing $\{\mathbf{H}'_i\}$ which can be classified into *compositional* and *additive* methods.

4.1. Compositional methods. These techniques are based on the composition of transformations. Depending on the reference system, we may define three different approaches: *compositional*, *global* and *local* smoothing approaches.

4.1.1. Compositional approach. If the camera is static, it may be sound to find and compensate for the homographies toward a fixed frame, or even a background image obtained by former registrations and accumulation. The compensating transformations are obtained through compositions from the current image to the reference frame. The compositional approach is extremely relevant for fixed cameras for which a reference background frame can be established.

Definition 4.1. *We define the* compositional *approach as the process of calculating the transformations that compensate the images with respect to a reference frame. The transformations are obtained by composing the relative motions between successive frames by*

$$\mathbf{H}'_i = \mathbf{H}_{1,i} := \prod_{j=2}^{N} \mathbf{H}_{j-1,j} = \mathbf{H}_{i-1,i}\mathbf{H}_{i-2,i-1}\cdots\mathbf{H}_{2,3}\mathbf{H}_{1,2}. \tag{4.1}$$

This is the technique used in the first works [21, 52]; it was usually combined with the

Figure 3. *Compositional approach and the problem of changing the focal length or moving the camera forwards/backwards. Left, the first frame of the sequence; middle, another frame of the sequence; right, the compensated image for that frame. The problem with the compositional approach is that there is a steady zoom in as the camera moves forward. Thus, compensating the motion by registering the frames up to the first one reduces the current frame more and more.*

construction of mosaics and is compatible with real-time processing, since each single frame can be directly compensated.

One of the drawbacks of this approach is that the composition of homographies also accumulates the errors introduced in the motion estimation process. If we choose the first image as the reference frame, this problem becomes more acute as the video goes on. One could avoid the accumulation of errors by registering the images directly to the initial reference frame. However, at some time, there may be no sufficient overlap between the objects in the current frame and the initial frame, or the illumination conditions may change.

If the camera moves, choosing a unique reference frame may introduce severe distortions. Nevertheless, there exist several satisfactory strategies in this case, such as the following:

(i) Choosing several reference frames equally distributed along the sequence, so that the rectification is carried out with respect to the nearest key frame. One of the drawbacks of this strategy is that this can create incongruous jumps in the video at the reference frames.

(ii) Computing homographies in both temporal directions using multiple reference frames. The frame in the middle of two key frames can be rectified using an average between the two composed homographies, from the left and right transformations. But this would force the result to maintain some frame position uncorrected.

(iii) Detect when the motion has gone beyond the range of the initial reference frame and pick another one [52].

Another shortcoming of the compositional approach is that it also compensates the intentional changes in focal length. In the same way, the method is not valid if the camera moves forwards or backwards. The undesirable effect is that the resolution of the images gets bigger or smaller, like in the example of Figure 3.

In fact, this problem appears when using similarities, affinities, or homographies for the compensation, because these include the scale factor, and transformations using (4.1) unduly compensate for the change of scale. The Euclidean transformations, or translations, work better in this setting, because they do not include the scale parameter.

4.1.2. Global smoothing approaches. Based on the previous approach, it is possible to design smoothing strategies that are suitable for moving cameras. The images are rectified

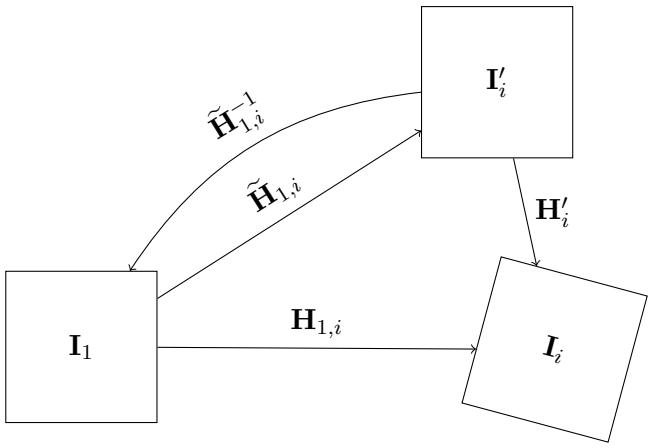

Figure 4. *Relation between the reference frame, $\mathbf{I}_1$, the actual frame, $\mathbf{I}_i$, and the compensated frame, $\mathbf{I}'_i$. This figure shows an intuitive way to compute the stabilizing transformation, $\mathbf{H}'_i$, from the estimated motion, $\mathbf{H}_{1,i}$, and the smoothed transformation, $\widetilde{\mathbf{H}}_{1,i}$.*

with respect to a fixed reference frame, but these techniques allow estimating a smooth path along the camera trajectory.

Compositional smoothing. The idea of compositional smoothing is illustrated in Figure 4. The objective is to find the transformations $\{\mathbf{H}'_i\}$ that relate $\{\mathbf{I}_i\}$ to $\{\mathbf{I}'_i\}$, so that the smooth transformations $\{\widetilde{\mathbf{H}}_{1,i}\}$ do not include the high frequency motion. The compositional smoothing approach simulates a smooth registration of the current frame with respect to a given reference frame.

Definition 4.2. *We start from the compositional transition homographies $\mathbf{H}_{1,i}$ defined by (4.1) and, more precisely, from their eight matrix elements $\mathbf{H}_{1,i}(p,q)$, where p and q are the row and column index, respectively. Consider the smooth homographies $\widetilde{\mathbf{H}}_{1,i}$, whose elements $\widetilde{\mathbf{H}}_{1,i}(p,q)$ are obtained through a convolution with a discrete Gaussian function of the series $\mathbf{H}_{1,i}(p,q)$ as*

$$\widetilde{\mathbf{H}}_{1,i}(p,q) := (G_\sigma * \mathbf{H}_{1,\cdot})_i(p,q) = \sum_{j\in\mathcal{N}_i} G_\sigma(j-i)\mathbf{H}_{1,j}(p,q), \tag{4.2}$$

with $G_\sigma(x) := We^{-\frac{x^2}{2\sigma^2}}$, $\mathcal{N}_i = \{j : i-k \le j \le i+k\}$, and $W := 1/\sum_{i=-k}^{i=k} e^{-\frac{i^2}{2\sigma^2}}$ a normalizing coefficient. The compositional smoothing *approach is defined by the following rectifying transformations and image stabilized sequence*

$$\mathbf{H}'_i := \mathbf{H}_{1,i}\widetilde{\mathbf{H}}^{-1}_{1,i}\ ; \qquad \mathbf{I}'_i(\mathbf{x}) := \mathbf{I}_i(\mathbf{H}'_i\mathbf{x}\}. \tag{4.3}$$

Note that we are using the compositions with respect to the first frame, and each element is obtained as a weighted average of the elements of the composed homographies. The next theorem verifies that this is a correct stabilization.

Theorem 4.3. *Given valid image deformations* $\mathbf{H}_{1,i}$ *from frame* i *to reference frame* 1*, such that* $\mathbf{I}_1(\mathbf{x}) = \mathbf{I}_i(\mathbf{H}_{1,i}\mathbf{x})$*, the smooth motion compensation for frame* i *defined by*

$$\mathbf{H}'_i := \mathbf{H}_{1,i}\widetilde{\mathbf{H}}_{1,i}^{-1} \tag{4.4}$$

is also a valid image deformation, as it satisfies

$$\mathbf{I}'_i(\mathbf{x}) =: \mathbf{I}_1(\widetilde{\mathbf{H}}_{1,i}^{-1}\mathbf{x}) = \mathbf{I}_i(\mathbf{H}_{1,i}\widetilde{\mathbf{H}}_{1,i}^{-1}\mathbf{x}). \tag{4.5}$$

Proof. To demonstrate this relation, note that

$$\begin{aligned} \mathbf{I}_1(\mathbf{x}) &= \mathbf{I}_i(\mathbf{H}_{1,i}\mathbf{x}), \\ \mathbf{I}_1(\mathbf{x}) &= \mathbf{I}'_i(\widetilde{\mathbf{H}}_{1,i}\mathbf{x}). \end{aligned} \tag{4.6}$$

Given that the transformations are invertible, we obtain from the second equation in (4.6)

$$\mathbf{I}'_i(\mathbf{x}) = \mathbf{I}_1(\widetilde{\mathbf{H}}_{1,i}^{-1}\mathbf{x}). \tag{4.7}$$

Using the first relation in (4.6), we have

$$\mathbf{I}'_i(\mathbf{x}) = \mathbf{I}_1(\widetilde{\mathbf{H}}_{1,i}^{-1}\mathbf{x}) = \mathbf{I}_i(\mathbf{H}_{1,i}\widetilde{\mathbf{H}}_{1,i}^{-1}\mathbf{x}). \tag{4.8}$$

■

Alternatively to Definition 4.2, the smoothing can be realized on the parameters of the transformations, $\mathbf{p}$, as detailed in Table 1. For instance, if we use a Euclidean transformation, the parameters to be smoothed are the translation, (t_x, t_y), and the angle of rotation, θ. Since they are treated separately, the smoothing has a geometrical meaning. However, for affinities and homographies, the smoothing is less intuitive and can be formalized with the transformation matrices. In the case of similarities, it is possible to use another parameterization based on the translation (t_x, t_y), the scale factor λ, and the angle of rotation θ. These can be easily obtained from the parameters in Table 1, and, again, the smoothing has a geometrical meaning.

The method explained in [52] follows this scheme, although it combines both the motion estimation and smoothing steps. Figure 5 shows the influence of σ in the stabilization. In this sequence, the camera moves forward and rotates 90º on the left. Choosing a small σ yields good results, but large values are not convenient. A good smoothing method should adapt the strength of the smoothing to the camera rotation velocity.

Boundary conditions for compositional smoothing. The Gaussian filtering of the sequence of homographies requires specifying how to deal with the temporal boundary conditions at the beginning and end of the time interval. When the smoothing radius goes beyond the limits of the image sequence, several strategies can be envisaged.

Definition 4.4. *Let* $\{\mathbf{H}_{1,i}\}$ *be a set of transformations from each frame* i *to the reference frame* 1 *in a time interval between* 1 *and* N*. We define* constant boundary conditions *by*

$$\mathbf{H}_{1,j} := \begin{cases} \mathbf{H}_{1,1} & \text{if } j < 1, \\ \mathbf{H}_{1,N} & \text{if } j > N. \end{cases} \tag{4.9}$$

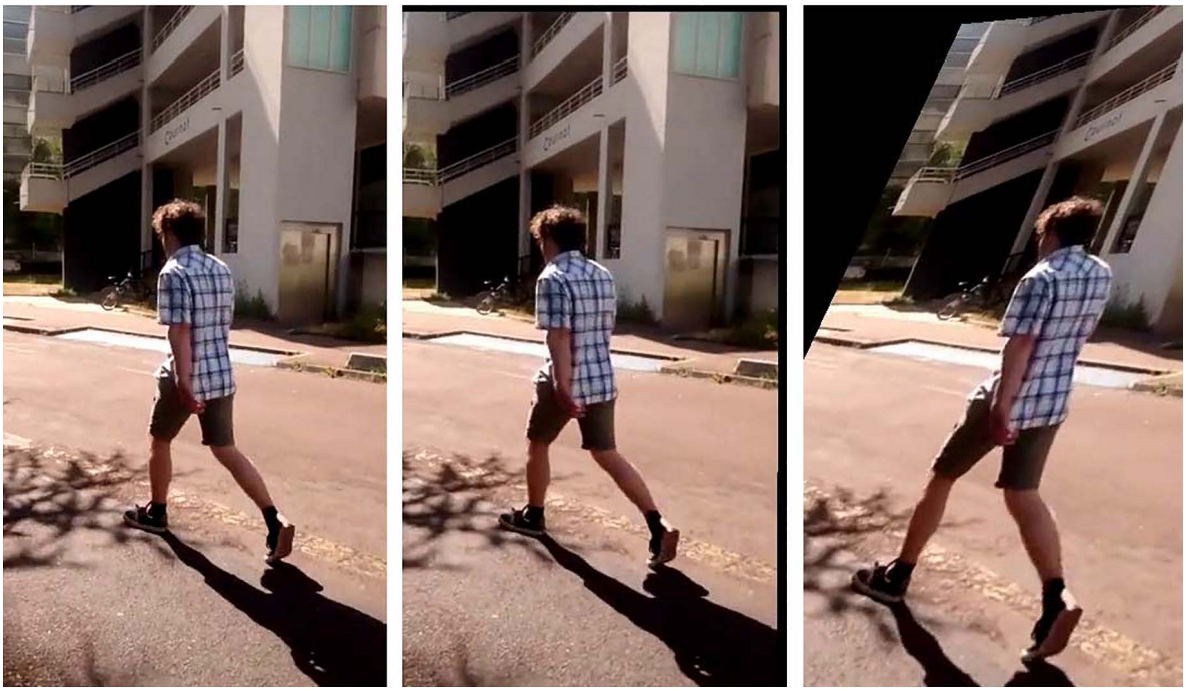

Figure 5. *Effect of exaggerated smoothing: Left image, one frame of the video; middle image, stabilization result with* $\sigma := 20$*; right image, stabilization with* $\sigma := 100$*. We used the direct method for motion estimation, homography transforms, and the local matrix–based smoothing strategy. In this movie the camera is rotating quickly, and smoothing in a large time interval is damaging.*

We define Neumann boundary conditions *in the range* $[-N+1, 2N-1]$ *by*

$$\mathbf{H}_{1,j} := \begin{cases} \mathbf{H}_{1,-j+1} & \text{if } j < 1, \\ \mathbf{H}_{1,2N-j} & \text{if } j > N. \end{cases} \tag{4.10}$$

We define Dirichlet boundary conditions *in the range* $[-2N+2, 3N-1]$ *by*

$$\mathbf{H}_{1,j} := \begin{cases} \mathbf{H}_{1,-j+2} + 2\mathbf{H}_{1,1} - 2\mathbf{H}_{1,N} & \text{if } -2N+2 \le j < -N+1, \\ 2\mathbf{H}_{1,1} - \mathbf{H}_{1,-j+1} & \text{if } -N+1 \le j < 1, \\ 2\mathbf{H}_{1,N} - \mathbf{H}_{1,2N-j} & \text{if } N \le j < 2N, \\ \mathbf{H}_{1,j-2N+1} + 2\mathbf{H}_{1,N} - 2\mathbf{H}_{1,1} & \text{if } 2N \le j < 3N. \end{cases} \tag{4.11}$$

Constant boundary conditions replicate the first and last transformations beyond the scope of the video. A consequence of this boundary condition is that, for large values of σ, the initial and final frames of the video are allowed to move from their original positions.

In the Neumann boundary conditions, the derivative of the homographies is constant in the boundaries. These conditions are accomplished by reflecting the values on both ends. Again, this allows the initial and final frames to move from their original positions.

The goal of Dirichlet boundary conditions is, by an odd reflection across the temporal boundaries, to ensure that the initial and final frames do not move, namely to obtain at the

boundaries the equivalent to the original matrices. With these boundary conditions, the first and last frames will coincide with the original video.

Compositional local smoothing. The *compositional local smoothing* approach is similar to the previous method, with the main difference being that the transformations are smoothed locally and then composed with the original transformations.

Definition 4.5. *Given the following convolution with a Gaussian function,*

$$\widetilde{\mathbf{H}}_{i,i+1}(p,q) := (G_\sigma * \{\mathbf{H}\})_i(p,q) = \sum_{j\in\mathcal{N}_i} G_\sigma(i-j)\mathbf{H}_{j,j+1}(p,q), \tag{4.12}$$

the compositional local smoothing *is defined by the rectifying transformations*

$$\mathbf{H}'_i := \prod_{j=1}^{i}\left(\mathbf{H}_{j,j+1}\widetilde{\mathbf{H}}^{-1}_{j,j+1}\right) = \left(\mathbf{H}_{i,i+1}\widetilde{\mathbf{H}}^{-1}_{i,i+1}\right)\left(\mathbf{H}_{i-1,i}\widetilde{\mathbf{H}}^{-1}_{i-1,i}\right)\cdots$$
$$\cdots\left(\mathbf{H}_{2,3}\widetilde{\mathbf{H}}^{-1}_{2,3}\right)\left(\mathbf{H}_{1,2}\widetilde{\mathbf{H}}^{-1}_{1,2}\right). \tag{4.13}$$

The stabilized image sequence is defined by (3.1), $\mathbf{I}'_i(\mathbf{x}) := \mathbf{I}_i(\mathbf{H}'_i\mathbf{x}\}$.

This relation is similar to the compositional smoothing approach in Figure 4, where the expression $\mathbf{H}_{i,i+1}\widetilde{\mathbf{H}}^{-1}_{i,i+1}$ removes the original local shake and introduces a smooth increment. This method is mentioned in [50].

Boundary conditions for compositional local smoothing. The boundary conditions for the compositional local smoothing approach are slightly different from the previous scheme, because the transformations are increments between consecutive images. In this case, we have the following conditions.

Definition 4.6. *Let* $\mathbf{Id}$ *be the identity matrix.*

Constant boundary conditions *are defined as* $\mathbf{H}_{j,j+1} := \mathbf{Id}$ *if* $j < 1$ *or* $j > N$.

Neumann boundary conditions *are defined in the range* $[-N+2, 2N-1]$ *as*

$$\mathbf{H}_{j,j+1} := \begin{cases} \mathbf{H}^{-1}_{-j+1,-j+2} & \text{if } j < 1, \\ \mathbf{H}^{-1}_{2N-j,2N-j+1} & \text{if } j > N. \end{cases} \tag{4.14}$$

Dirichlet boundary conditions *in the range* $[-N+1, 2N-1]$ *are defined by*

$$\mathbf{H}_{j,j+1} := \begin{cases} \mathbf{H}_{-j,-j+1} & \text{if } j < 1, \\ \mathbf{H}_{2N-j,2N-j+1} & \text{if } j > N. \end{cases} \tag{4.15}$$

4.1.3. Local smoothing approaches. The main problem with the previous approaches is that the composition of transformations from a given frame successively accumulates errors. Local methods, on the other hand, do not rely on a fixed reference image, but the coordinate system is centered at each frame independently. The smoothing is carried out in a temporal window centered at the frame.

We envisage two alternatives: in the first, the matrices are referenced to the current frame, and the components of the matrices are smoothed similarly to the previous approaches; in the second strategy, a set of points is selected in the current frame, and their positions are tracked in the neighboring frames by applying the transition transforms.

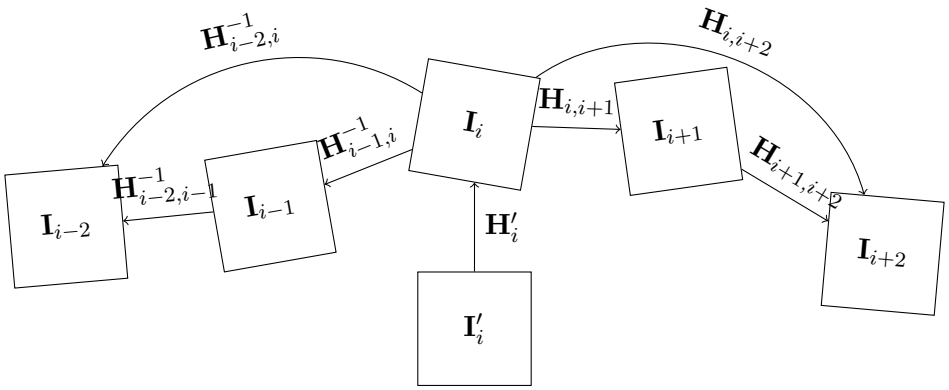

Figure 6. *Local matrix–based smoothing. The stabilizing transformation, $\mathbf{H}_i'$, is obtained from a local neighborhood of frame $\mathbf{I}_i$ using forward, $\mathbf{H}_{i,j}$, and backward, $\mathbf{H}_{i,j}^{-1}$, transformations.*

Local matrix–based smoothing. This method was proposed in [50] and is presented in Figure 6. The reference system is centered in the current frame, and the smoothing is carried out with a set of transformations around a temporal neighborhood. These transformations need to be related to this central frame, so the composition is given in both temporal directions.

Definition 4.7. *Consider the following compositions with the previous transformations from the current frame,*

$$\mathbf{H}_{i,j:j<i} = \prod_{l=j}^{i-1} \mathbf{H}_{l+1,l} = \mathbf{H}_{j+1,j}\mathbf{H}_{j+2,j+1}\cdots\mathbf{H}_{i-1,i-2}\mathbf{H}_{i,i-1}, \tag{4.16}$$

with $\mathbf{H}_{l+1,l} = \mathbf{H}_{l,l+1}^{-1}$, and the composition with the following frames as

$$\mathbf{H}_{i,j:j>i} = \prod_{l=i}^{j} \mathbf{H}_{l,l+1} = \mathbf{H}_{j-1,j}\mathbf{H}_{j-2,j-1}\cdots\mathbf{H}_{i-1,i}\mathbf{H}_{i,i+1} \tag{4.17}$$

in a neighborhood, $\mathcal{N}_i = \{j : i-k \leq j \leq i+k\}$, around frame i. Define the convolution with a Gaussian function in this interval by

$$\widehat{\mathbf{H}}_i(p,q) := \sum_{j=i-k}^{i+k} G_\sigma(i-j)\mathbf{H}_{i,j}(p,q), \tag{4.18}$$

with $\mathbf{H}_{i,i} := \mathbf{Id}$. Then the local matrix–based *stabilization is defined by the rectifying transformations $\mathbf{H}_i' := \widehat{\mathbf{H}}_i^{-1}$ and the stabilized image sequence by (3.1), $\mathbf{I}_i'(\mathbf{x}) := \mathbf{I}_i(\mathbf{H}_i'\mathbf{x})$.*

The boundary conditions for this approach are identical to those for the compositional smoothing approach, so that Definition 4.4 is valid. This approach is more stable and works better for more complex transformations, such as affinities and homographies.

One of the clear benefits of this approach is that the influence of errors, which may have been produced during the motion estimation process, is limited to a region given by the

temporal window in (4.18). As a consequence, the method is more stable and may recover from errors during the compensation. In the previous techniques, once an error is introduced in the motion of an image, it is propagated to the rest of frames.

This idea is also used in [56] for the smoothing of 3D camera rotation matrices. In that case, it also compensates for the rolling shutter effects.

Local point–based smoothing. This variant proposes a more intuitive way to process the video by smoothing point trajectories instead of matrices. The idea is to select several points in the current frame and track them in its neighborhood using the estimated homographies. A Gaussian convolution in each path will provide an average set of points. Then, the stabilization homographies can be computed from the averaged points.

If we want to compute a homography—eight parameters—we need to select at least four points in each frame, $\{\mathbf{x}_i^p\}^{p=1,\dots,4}$. These points are then projected forwards and backwards as $\mathbf{x}_{i+1}^p := \mathbf{H}_{i,i+1}\mathbf{x}_i^p$ and $\mathbf{x}_{i-1}^p := \mathbf{H}_{i-1,i}^{-1}\mathbf{x}_i^p$, respectively.

Definition 4.8. *Consider a set of points in the current frame i, $\{\mathbf{x}_i^p\}^{p=1,\dots,N}$, and a trajectory for each point in a temporal neighborhood, $\mathcal{N}_i = \{j : i-k \leq j \leq i+k\}$, given by the following points:*

$$\text{for } j = i-k,\dots,i-1,\quad \mathbf{x}_j^p := \left(\prod_{l=j}^{i-1}\mathbf{H}_{l+1,l}\right)\mathbf{x}_i^p = \mathbf{H}_{j+1,j}\mathbf{H}_{j+2,j+1}\cdots \cdots\mathbf{H}_{i-1,i-2}\mathbf{H}_{i,i-1}\mathbf{x}_i^p \tag{4.19}$$

on the left, and

$$\text{for } j = i+1,\dots,i+k,\quad \mathbf{x}_j^p := \left(\prod_{l=i}^{j-1}\mathbf{H}_{l,l+1}\right)\mathbf{x}_i^p = \mathbf{H}_{j-1,j}\mathbf{H}_{j-2,j-1}\cdots \cdots\mathbf{H}_{i+1,i+2}\mathbf{H}_{i,i+1}\mathbf{x}_i^p \tag{4.20}$$

on the right. We define the convolution of each trajectory with a Gaussian function by

$$\text{for } p = 1,\dots,N,\quad \widetilde{\mathbf{x}}_i^p := \left(G_\sigma * \{\mathbf{x}_j^p\}_{j=i-k,\dots,i+k}\right)_i. \tag{4.21}$$

The local point–based smoothing *approach is then obtained by the rectifying transformations $\mathbf{H}_i' := \widehat{\mathbf{H}}_i^{-1}$, where $\widehat{\mathbf{H}}_i$ is calculated from points $\{\mathbf{x}_i^p\}$ to the smoothed set $\{\widetilde{\mathbf{x}}_i^p\}$. Finally, the stabilized image sequence is obtained by (3.1) as $\mathbf{I}_i'(\mathbf{x}) := \mathbf{I}_i(\mathbf{H}_i'\mathbf{x})$.*

In order to compute the homography from the points, we may follow the strategies proposed in [23]. The inhomogeneous system, where one element of the matrix is set constant, is the simplest one.

The number of points to track depends on the type of transformation we choose. Since every point establishes two equations, we need one point for translations, two for Euclidean and similarity transforms, three for affinities, and four for homographies. For Euclidean transformations there is one spare equation, and we have an overdetermined system. Another

alternative is to always track four points and then adapt the resulting homography to the chosen parameterization.

Here, logically, the boundary conditions are applied to the points themselves with a definition similar to the one applied on the local matrix–based smoothing.

Definition 4.9. *Given a point* $\mathbf{x}_i^p$ *in the reference frame:*
Constant boundary conditions *are defined by*

$$\mathbf{x}_j^p := \begin{cases} \mathbf{x}_1^p & \text{if } j < 1, \\ \mathbf{x}_N^p & \text{if } j > N. \end{cases} \tag{4.22}$$

Neumann boundary conditions *are defined in the range* $[-N+1, 2N-1]$ *by*

$$\mathbf{x}_j^p := \begin{cases} \mathbf{x}_{-j+1}^p & \text{if } j < 1, \\ \mathbf{x}_{2N-j}^p & \text{if } j > N. \end{cases} \tag{4.23}$$

Dirichlet boundary conditions *are defined in the range* $[-2N+2, 3N-1]$ *by*

$$\mathbf{x}_j^p := \begin{cases} \mathbf{x}_{-j+2}^p + 2\mathbf{x}_1^p - 2\mathbf{x}_N^p & \text{if } -2N+2 \le j < -N+1, \\ 2\mathbf{x}_1^p - \mathbf{x}_{-j+1}^p & \text{if } -N+1 \le j < 1, \\ 2\mathbf{x}_N^p - \mathbf{x}_{2N-j}^p & \text{if } N \le j < 2N, \\ \mathbf{x}_{j-2N}^p + 2\mathbf{x}_N^p - 2\mathbf{x}_1^p & \text{if } 2N \le j < 3N. \end{cases} \tag{4.24}$$

4.2. Additive methods. Instead of compositions, these techniques rely on *virtual trajectories* obtained by time integration of apparent frame to frame motions. The benefit of these schemes is that the errors produced by the compositions are not accumulated. We call these approaches *additive methods* because the information is computed in an incremental way through the addition of the transformations.

In this group, we shall distinguish two approaches: The first is based on the integration of the coefficients of the transformations and the second on the integration of the local motion of fixed points in the video frames.

Local linear matrix–based smoothing. This technique proposes a linear variant of the local matrix–based smoothing.

Definition 4.10. *We define the virtual trajectory of a homography as*

$$\bar{\mathbf{H}}_i := \mathbf{Id} + \sum_{l=1}^{i-1} (\mathbf{H}_{l,l+1} - \mathbf{Id}) . \tag{4.25}$$

The smoothed coefficients of the matrix trajectory for $i = 1, \dots, N$ *are defined by*

$$\widetilde{\mathbf{H}}_i(p,q) := \left(G_\sigma * \{\bar{\mathbf{H}}_j(p,q)\}\right)_i = \sum_{j=-k}^{j=k} G_\sigma(j) \bar{\mathbf{H}}_{i-j}(p,q) \tag{4.26}$$

and

$$\widehat{\mathbf{H}}_i = \widetilde{\mathbf{H}}_i - \bar{\mathbf{H}}_i + \mathbf{Id}. \tag{4.27}$$

The local linear matrix–based smoothing *approach is then obtained by the rectifying transformations* $\mathbf{H}'_i := \widehat{\mathbf{H}}_i^{-1}$. *Finally, the stabilized image sequence is obtained by* (3.1) *as* $\mathbf{I}'_i(\mathbf{x}) := \mathbf{I}_i(\mathbf{H}'_i\mathbf{x}\}$.

The boundary conditions for this method are equivalent to the local matrix–based smoothing strategy, as given in Definition 4.4.

Since we obtain a unique virtual trajectory for the whole sequence, it is easy to apply a discrete cosine transform (DCT) based Gaussian filtering. This method is preferred to discrete filters because it is more precise [16].

In the case of a pure translation, linear matrix–based smoothing is equivalent to local linear point–based smoothing, which we shall examine next. However, for general transformations, the matrix-based scheme is less sensitive to abrupt camera changes and numerical inaccuracies.

Local linear point–based smoothing. This variant proposes a linear system to process the video by smoothing point trajectories. The *virtual trajectories*, $\{\mathbf{x}_i^p\}^{p=1,\dots,4}$, of the reference points are computed by integrating their temporal discrete derivative. Then, as in the preceding section, they are smoothed out to compute a new position at time i. The new position of these points yields the stabilization homography.

Definition 4.11. *Consider a fixed set of points in frame coordinates,* $\{\mathbf{x}^p\}^{p=1,\dots,P}$, *and define their virtual trajectory by* $\mathbf{x}_1^p = \mathbf{x}^p$ *and*

$$\text{for } i = 1,\dots,N-1, \quad \mathbf{x}_i^p := \mathbf{x}^p + \sum_{l=1}^{i-1} (\mathbf{H}_{l,l+1}\mathbf{x}^p - \mathbf{x}^p)\,. \tag{4.28}$$

We define the stabilized position of $\mathbf{x}_i^p$ *by*

$$\widetilde{\mathbf{x}}_i^p := \left(G_\sigma * \{\mathbf{x}_j^p\}\right)_i = \sum_{j=-k}^{j=k} G_\sigma(j)\mathbf{x}_{i-j}^p, \tag{4.29}$$

$$\widehat{\mathbf{x}}_i^p := \mathbf{x}^p - \mathbf{x}_i^p + \widetilde{\mathbf{x}}_i^p. \tag{4.30}$$

The local linear point–based smoothing *approach is then obtained by the rectifying transformations* $\mathbf{H}'_i := \widehat{\mathbf{H}}_i^{-1}$, *where the homography* $\widehat{\mathbf{H}}_i$ *is calculated from the fixed points* $\{\mathbf{x}^p\}$ *to the stabilized set at frame* i, $\{\widehat{\mathbf{x}}_i^p\}$. *Finally, the stabilized image sequence is obtained by* (3.1) *as* $\mathbf{I}'_i(\mathbf{x}) := \mathbf{I}_i(\mathbf{H}'_i\mathbf{x}\}$.

The boundary conditions for this method are equivalent to the local point–based smoothing strategy, as given in Definition 4.9.

A clear benefit of the additive approaches is that, unlike the previous techniques, no composition between homographies is performed. This reduces the effect of numerical errors or the influence of errors during the motion estimation step. Another advantage is that it is easy to use a Gaussian convolution based on the DCT, since we obtain a unique 1D signal for each point trajectory or matrix coefficient.

In the experimental results, we will use a linear scale-space analysis which is based on a formulation similar to (4.28). Furthermore, we will see that these two approaches behave much better if we analyze the empty regions created by the stabilization.

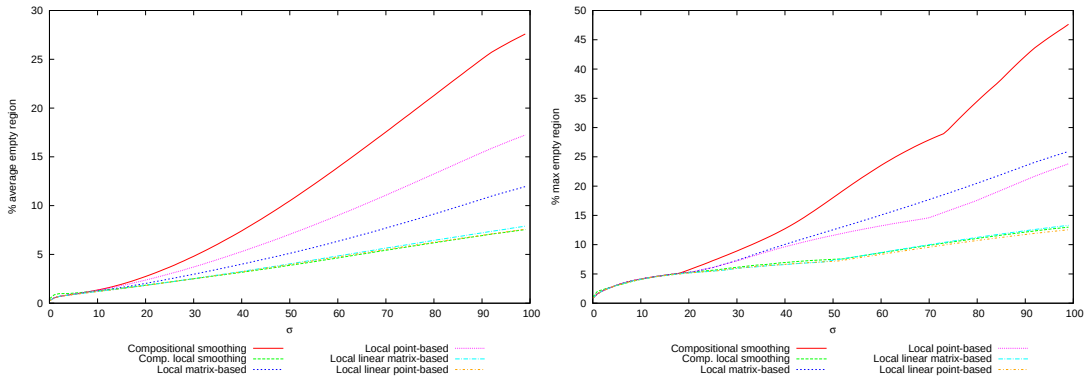

Figure 7. *Office: Average (left) and maximal (right) percentage of empty regions as a function of* σ.

5. Which smoothing strategy is the best? A postprocessing step is necessary to remove the empty regions that appear at the border of the images. The simplest approach, and probably the most used, is Crop & Zoom. The idea is to find the largest area axis-parallel rectangle [13] that does not contain empty regions and to apply the adequate crop to all frames to remove them. This can be realized by translating and scaling the rectangle in order to fit inside the image dimensions. This process can be executed before the warping step in the smoothed homographies to account for this similarity transformation.

The necessity of such a process after motion stabilization gives also a natural criterion to compare stabilization strategies. Cropping must be maximized because it implies either a resolution loss by digital zoom-in if the video frame size is maintained, or a reduction of this size.

A natural criterion to compare the motion compensation strategies is to measure the empty space produced after compensating camera motion. Working on a series of realistic videos, we explored the average and maximum empty spaces created by every strategy. The rationale of this comparison is that large empty regions provide poor solutions and that a smoothing strategy will be preferred if it corrects video shake with minimal information loss.

Looking at Figure 7, we observe that all the strategies have similar behavior for small values of σ. However, the compositional smoothing approach is more sensitive to large values. This figure shows the average and maximum percentage of empty regions produced by each strategy on the *Office* video, a video containing camera shake and a strong continuous rotation. The local matrix– and local point–based smoothing approaches behave better for large smoothing values, with the matrix-based technique having a better average graphic. The performance of the maximum percentage of empty regions for the local point–based technique is slightly better. The best approaches are the local linear–based methods, with very similar behaviors. Paradoxically, the compositional local smoothing approach provides similar results in this sequence. These results can also be observed in the last two columns of Table 4.

Table 2 shows the average and maximum percentages of empty regions for the brutal video obtained by a GoPro fixed to the chest of a running man. This example illustrates the limits of smoothing strategies in such extreme cases. The best strategy, which turns out to be local

Table 2

Man running with a GoPro in his chest: Average and maximum percentages of empty regions in the border of the compensated frames ($\sigma := 30$).

Method	average	maximum
Compositional	94%	100%
Compositional smoothing	27,9%	100%
Compositional local smoothing	23,6%	46,6%
Local matrix–based smoothing	20,5%	100%
Local point–based smoothing	52,4%	100%
Local linear matrix–based smoothing	17,1%	35,3%
Local linear point–based smoothing	**16,8**%	**35,2**%

Table 3

Average percentage of empty regions left on the border of the compensated frames for ($\sigma := 30$). This percentage is given for each kind of transform, from translation to homography, and for each kind of smoothing process, from compositional to local linear point–based. This criterion illustrates two phenomena: First, the growing numbers from left to right show that allowing more degrees of freedom to the transform implies also a more severe correction, as is logical; second, the table is a clear decider about the preferable smoothing process. The local linear matrix–based and the (similar) local linear point–based processes win as they guarantee the smallest distortion and the smallest distortion increase when increasing the number of degrees of freedom.

Method	Translat.	Euclid.	Similar.	Affine	Homography
Compositional	20,5%	23,8%	21,7%	23,2%	43,8%
Compositional smoothing	**4,0**%	**4,1**%	6,3%	6,4%	8,4%
Compositional local smth.	8,3%	10,5%	8,1%	8,5%	18,9%
Local matrix–based	**4,0**%	**4,0**%	6,3%	6,4%	6,1%
Local point–based	**4,0**%	**4,1**%	6,7%	6,8%	11,2%
Local linear matrix–based	**4,0**%	**4,1**%	**4,6**%	**4,6**%	**5,5**%
Local linear point–based	**4,0**%	**4,0**%	**4,6**%	**4,6**%	**5,9**%

linear point–based smoothing, loses 16,8% of the frame on average, similar to the local linear matrix–based smoothing. The maximum loss in these cases is 35,2% and 35,3%, respectively, which is much better than those of the compositional techniques.

Table 3 presents the average image percentage for the empty regions left on the border of the compensated frames for moderate smoothing ($\sigma := 30$) and ten diverse videos. This average was computed for each kind of geometric transform, from translation to homography, and for all kinds of smoothing processes that we have considered, from compositional to local linear point–based with Neumann boundary condition.

A lower empty region percentage is a logical and factual criterion of success: For a fixed Gaussian smoothing, it measures the loss of resolution of the video caused by the camera shake correction. Clearly large percentages like 23% would not be acceptable. Fortunately, the best methods give a reasonable 5 to 6%.

The results demonstrate two phenomena: First, the increasing numbers from left to right on each row show that allowing more degrees of freedom in the transform implies also a more severe correction; second, the table is a clear decider about the preferable smoothing process. The local linear matrix–based and the (similar) local linear point–based processes win as they guarantee the smallest distortion and the smallest distortion increase when increasing the number of degrees of freedom. Table 5 shows the average percentage of empty regions left

Table 4

Average and maximum percentages of empty regions in the border of the compensated frames after stabilization for five videos with different dynamics: Descending stairs, Man running with a camera on his head, Earthquake, Cournot Building, and Office, respectively. In all cases, we choose $\sigma := 30$. C stands for compositional, CS for compositional smoothing, CLS for compositional local smoothing, LMS for local matrix–based smoothing, LPS for local point–based smoothing, LLM for local linear matrix–based smoothing, and LLP for local linear point–based smoothing.

Method	Desc. stairs		Man running		Earthquake		Building		Office	
	mean	max	mean	max	mean	max	mean	max	mean	max
C	30,5%	63,4%	81,0%	100%	26,0%	62,1%	2,6%	7,9%	64,6%	100,0%
CS	**4,3%**	**12,6%**	**4,2%**	27,4%	**1,7%**	9,0%	**0,4%**	**2,1%**	4,6%	8,6%
CLS	5,5%	13,5%	10,9%	**14,3%**	2,5%	9,9%	1,2%	**2,0%**	**2,5%**	6,1%
LMS	**4,4%**	**12,6%**	**4,0%**	27,4%	**1,8%**	9,4%	**0,4%**	**2,1%**	2,9%	7,1%
LPS	5,2%	13,5%	18,0%	31,2%	3,4%	59,5%	**0,4%**	**1,6%**	3,6%	7,0%
LLM	**4,3%**	**12,6%**	**4,9%**	27,7%	**1,8%**	**8,3%**	**0,4%**	**2,1%**	**2,5%**	**5,8%**
LLP	**4,2%**	**12,5%**	**4,7%**	**21,8%**	**1,7%**	**8,4%**	**0,4%**	**2,1%**	**2,4%**	**5,9%**

Table 5

Average percentage of the empty regions left on the border of the compensated frames for ($\sigma := 60$). This percentage is given for each kind of transform, from translation to homography, and for each kind of smoothing process, from compositional to local linear point–based. This table confirms, for more drastic smoothing, the conclusion given for Table 3: The local linear matrix–based and the (similar) local linear point–based processes win. They guarantee the smallest distortion and the smallest distortion increase when increasing the number of degrees of freedom. Interestingly, the winners are also the closest ones to the linear scale space strategy designed for the video analysis.

Method	Translat.	Euclid.	Similar.	Affine	Homography
Compositional	20,5%	23,8%	21,7%	23,2%	43,8%
Compositional smoothing	**5,5%**	**5,6%**	12,6%	12,7%	13,5%
Compositional local smth.	11,9%	14,2%	10,6%	10,5%	21,7%
Local matrix–based	**5,5%**	**5,5%**	12,6%	12,7%	11,0%
Local point–based	**5,5%**	**5,5%**	14,0%	14,1%	18,3%
Local linear matrix–based	**5,5%**	**5,5%**	**7,3%**	**7,3%**	**8,9%**
Local linear point–based	**5,5%**	**5,6%**	**7,3%**	**7,3%**	**9,4%**

on the border of the compensated frames for strong smoothing ($\sigma := 60$). This percentage is again given for each kind of transform, from translation to homography, and for each kind of smoothing process, from compositional to local linear point–based.

This table confirms, for more drastic smoothing, the conclusion given for Table 3: The local linear matrix–based and the (similar) local linear point–based processes win. They guarantee the smallest distortion and the smallest distortion increase when increasing the number of degrees of freedom. Interestingly, the winners are also the closest ones to the linear scale space strategy designed for video analysis in the next section.

The tables also confirm that the local matrix–based approach is the third best method and it is, on average, the best of the compositional approaches, as shown in [50]. We notice that, although the compositional and compositional local smoothing approaches provide good results for fewer degrees of freedom, and may yield good results for some sequences, they usually provide worse results for large image sequences, as can be seen in the first experiment in the next section.

Table 4 gives a detailed comparative view of the results for several videos. The table shows the average and maximum percentages of empty regions for each smoothing technique and for five very different videos. From left to right, the results are shown for *Descending stairs*, a camera carried backward by a cameraman; *Man running with a camera on his head*, a brisk running video; *Earthquake*, the violent vibration of a static camera on a pole; *Cournot Building*, which is the result of a camera phone held in a hand in front of a static scene; and *Office*, which is the result of a hand-held camera rotating strongly in front of a static scene. A perusal of these examples shows that the linear strategies are the best performing methods, followed by the local matrix–based and the compositional smoothing approaches.

The supplementary material includes three videos. The first two (M112715_01.mp4 [local/web 1.37MB] and M112715_02.mp4 [local/web 1.66MB]) compare the stabilization using the compositional local and the local matrix–based smoothing strategies. The result of the latter is more satisfactory because it produces fewer empty regions at the borders of the video frames. It shows that a global technique typically introduces more errors than a local approach. The third video (M112715_03.mp4 [local/web 1.12MB]) shows a comparison of Neumann and Dirichlet boundary conditions. We observe that, for Dirichlet conditions, the initial and final frames of the stabilized video coincide with the original sequence.

6. The motion linear scale-space.

6.1. The formal scale-space definition.

If $\mathbf{H}_{i,i+1}$ are the original transformations between consecutive frames and $\mathbf{H}'_i$ the rectifying homographies, we calculate the set of transformations for the stabilized sequence as

$$\widetilde{\mathbf{H}}_{i,i+1} := (\mathbf{H}'_{i+1})^{-1}\mathbf{H}_{i,i+1}\mathbf{H}'_i, \tag{6.1}$$

with $\widetilde{\mathbf{H}}_{0,1} := \mathbf{Id}$. These are used for computing the trajectories and scale-space graphics in the experiments. Our goal in this section is to define sets of motion signals that can undergo a multiscale analysis and that give a reliable geometric account of the camera motion. We assume that a series of homographies $\{\mathbf{H}_i\}_{i=1,\dots,N}$ between successive frames is given. These may be the initial homographies or the smoothed ones. We want to define and visualize representative *virtual trajectories* associated with these series. This virtual trajectory is computed as the integral of the instantaneous velocity of a fixed frame point. It is obtained by applying the transition transform of the current frame to this point. Its instantaneous velocity is defined as the difference between this transformed point and the fixed point in the frame. Integrating this instantaneous velocity yields a *virtual trajectory of the central point.*

Definition 6.1. *Let* $\mathbf{x}$ *be a fixed point in the video frame domain (for example, its central point). Given a series* $\{\mathbf{H}_i\}_{i=1,\dots,N}$ *of homographies between successive frames, we call the* virtual trajectory of $\mathbf{x}$ *the series*

$$\mathbf{x}_i := \mathbf{x} + \sum_{j=1}^{j=i-1} (\mathbf{H}_i\mathbf{x} - \mathbf{x}). \tag{6.2}$$

We call the instantaneous zoom *the area ratio between the quadrangles* $(\mathbf{H}_i\mathbf{w}, \mathbf{H}_i\mathbf{x}, \mathbf{H}_i\mathbf{y}, \mathbf{H}_i\mathbf{z})$ *and* $(\mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z})$*, where the second quadrangle is the image domain. We call the* instantaneous

rotation *of the video the angle between* $(\mathbf{H}_i\mathbf{u}, \mathbf{H}_i\mathbf{v})$ *and* $(\mathbf{u}, \mathbf{v})$*, where* $(\mathbf{u}, \mathbf{v})$ *denote the middle points of the left and right sides of the video frame.*

Our scale-space analysis receives discrete signals (virtual trajectories) obtained for the series of transition homographies. The trajectories are computed by Definition 6.1. These trajectories will be smoothed by Gaussian convolutions to preserve all the typical scale properties, namely translation invariance, locality, and causality. Thus we need first to specify how the Gaussian definition must be done to preserve these three properties.

Consider a digital signal $\mathbf{u}_k$ with $k = 0, \dots, N-1$ and its Discrete Fourier Transform (DFT) interpolate

$$u(x) = \sum_{m \in [-N/2, N/2-1]} \tilde{u}_m e^{\frac{2i\pi m x}{N}}, \tag{6.3}$$

where $\tilde{u}_m$ are the DFT coefficients of the N samples $\mathbf{u}_k$. The classic image analysis theory called "scale-space" convolves u with a Gaussian $G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$. A direct calculation shows that the result of the convolution of u with G_σ is

$$v(x) := (G_\sigma * u)(x) = \sum_{m \in [-\frac{N}{2}, \frac{N}{2}-1]} \tilde{u}_m \hat{G_\sigma}\left(\frac{2m\pi}{N}\right) e^{\frac{2i\pi m.x}{N}} \tag{6.4}$$

with $\hat{G}_\sigma(\xi) = e^{\frac{-\sigma^2\xi^2}{2}}$. Indeed,

$$G_\sigma * e^{\frac{2i\pi m.x}{N}} = \int_{\mathbf{R}} G_\sigma(y) e^{\frac{2i\pi m.(x-y)}{N}} dy = e^{\frac{2i\pi m.x}{N}} \int_{\mathbf{R}} G_\sigma(y) e^{-\frac{2i\pi m.y}{N}} dy = e^{\frac{2i\pi m.x}{N}} \hat{G_\sigma}\left(\frac{2\pi m}{N}\right).$$

The convolved discrete digital signal therefore simply is

$$\mathbf{v}_k := G_\sigma * \mathbf{u}_k = \sum_{m \in [-\frac{N}{2}, \frac{N}{2}-1]} \hat{G_\sigma}\left(\frac{2m\pi}{N}\right) \tilde{u}_m e^{\frac{2i\pi m.k}{N}}. \tag{6.5}$$

These calculations lead us to the following definition of Gaussian filtering of a discrete motion signal.

Definition 6.2. *Given a signal* $\mathbf{u}_k$*,* $k = 0, \dots, N-1$*, we call the DFT convolution of this signal by a Gaussian* G_σ *the discrete signal* $G_\sigma * \mathbf{u}_k$ *obtained by* (6.5)*.*

Nevertheless this definition implicitly assumes the signal to be N-periodic, which is generally not adequate, as it introduces artificial discontinuities at both ends of the signal. For this reason, the convolution must respect the constant, Neumann, or Dirichlet boundary conditions used in all our previous definitions of motion smoothing. This leads to the obvious next definition, which proposes again extending the signal adequately on an interval of length $3N$ before applying the DFT convolution.

Definition 6.3. *Given a signal* $\{\mathbf{v}_k\}$ *and a standard deviation for the Gaussian* σ*, we set for* $k = N$ *to* $2N-1$*,* $\mathbf{va}_k = \mathbf{v}_{k-N+1}$*.*

(i) *For a constant boundary condition, for* $k = 1$ *to* $N - 1$, $\mathbf{va}_k = \mathbf{v}_1$, *and for* $k = 2N$ *to* $3N - 2$, $\mathbf{va}_k = \mathbf{v}_N$.

(ii) *For a Neumann boundary condition, for* $k = 1$ *to* $N - 1$, $\mathbf{va}_k = \mathbf{v}_{N-k-1}$, *and for* $k = 2N$ *to* $3N - 2$, $\mathbf{va}_k = \mathbf{v}_{3N-k-1}$.

(iii) *For a Dirichlet boundary condition, for* $k = 1$ *to* $N - 1$, $\mathbf{va}_k = 2\mathbf{v}_1 - \mathbf{v}_{N-k-1}$, *and for* $k = 2N$ *to* $3N - 2$, $\mathbf{va}_k = \mathbf{v}_N - \mathbf{v}_{3N-k-1}$.

Then apply the DFT convolution of the signal $\mathbf{va}_k$ *with the Gaussian* G_σ *by Definition 6.2 to obtain a discrete signal* $\mathbf{sv}_k$, $k = 1, \ldots, 3N - 2$. *Finally, extract the meaningful part of the signal for* $k = 1$ *to* N, *by setting* $\mathbf{sv}_k = \mathbf{vb}_{N+k-1}$.

Our definition of the motion scale-space will be applied to four signals: the virtual trajectory of the central point, the instantaneous zoom, and the instantaneous rotation.

Definition 6.4. *Let* $\mathbf{u}_k$, $k = 1, \ldots, N - 1$, *be one of the movie motion signals given by Definition 6.1. We call the* scale-space *of* $\mathbf{u}_k$ *the following series of signals, which are also displayed in this order from bottom to top in all graphical representations:*

(a) *the original virtual trajectory* $\mathbf{u}_k$,

(b) *the high pass filtered signal* $\mathbf{v_{10}} := \mathbf{u}_k - (G_{10} * \mathbf{u})_k$ *where* $G_{10} * \mathbf{u}$ *is as defined in Definition 6.3*,

(c) *the low passed version of* (b) *defined by* $\mathbf{v}_{20} := (G_{10} * \mathbf{u})_k - (G_{20} * \mathbf{u})_k$,

(d) *the low passed version of* (c) *defined by* $\mathbf{v}_{40} := (G_{20} * \mathbf{u})_k - (G_{40} * \mathbf{u})_k$,

(e) *the low passed version of* (d) *defined by* $\mathbf{v}_{80} := (G_{40} * \mathbf{u})_k - (G_{80} * \mathbf{u})_k$,

(f) *the final low passed version of* $\mathbf{u}$, $\mathbf{u}_{80} = (G_{80} * \mathbf{u})_k$.

These definitions define a pyramid that can be collapsed back as we have $\mathbf{u} = \mathbf{u}_{80} + \mathbf{v}_{80} + \mathbf{v}_{40} + \mathbf{v}_{20} + \mathbf{v}_{10}$. Of course the basis value of $\sigma = 10$ can be replaced by any other. In addition, our scale-space contemplates the frequency analysis of the pyramid by DFT, namely the DFTs of $\mathbf{v}_{80}$, $\mathbf{v}_{40}$, $\mathbf{v}_{20}$, and $\mathbf{v}_{10}$, presented on the right of all graphics from top to bottom.

6.2. Scale-space experiments. We shall detail a first experiment, *Walking*, on a 5.400 frame video obtained from a camera hanging from the chest of a walking person. Figure 8 shows the results of all considered smoothing strategies. The compositional approach cannot follow the camera path because there are rotations larger than 90°. In the solution of the compositional local smoothing approach, there are more empty regions at the border of the image than in the local matrix–based smoothing strategy. This technique typically provides better results and is more stable for homographic transformations.

Figure 9 shows the *compositional trajectories* of the original and smoothed transformations. The compositional trajectories are obtained by successively composing the homographies to build a trajectory of the central pixel in the first frame. This sort of trajectory is different from the *virtual trajectory* in Definition 6.1, which will be used for the scale-space analysis. Observe that the compositional approach (green line) does not follow the trajectory of the camera. Most of the strategies show a similar behavior, except the compositional local smoothing approach at the end of the signal, especially in the y component.

We now pass to the scale-space analysis, for which we use Definition 6.1 for the virtual rotation and zoom trajectories and Definition 6.4 for the scale-space. The linear scale-space of Figure 10 is obtained by directly smoothing the virtual trajectory of the central point.

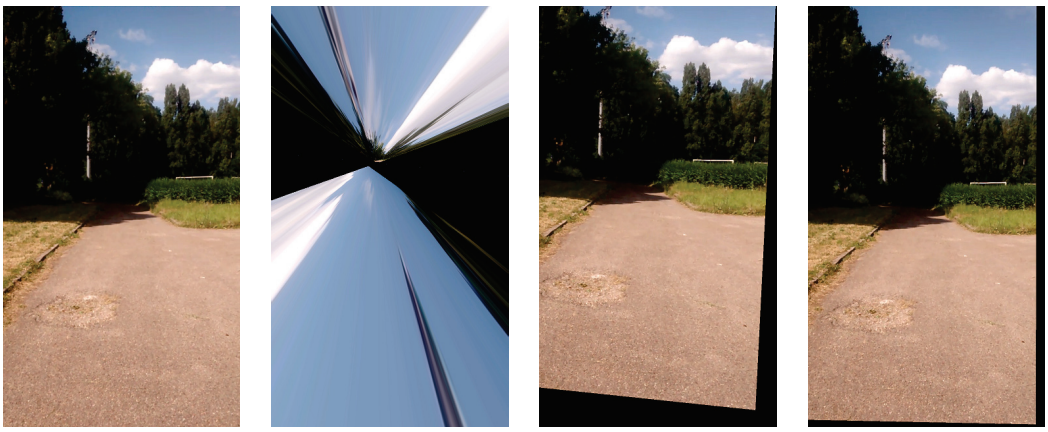

Figure 8. *Walking. From left to right: Image of the original sequence (frame* 800*) and the results of the compositional smoothing, the compositional local smoothing, and the local matrix–based smoothing approaches.*

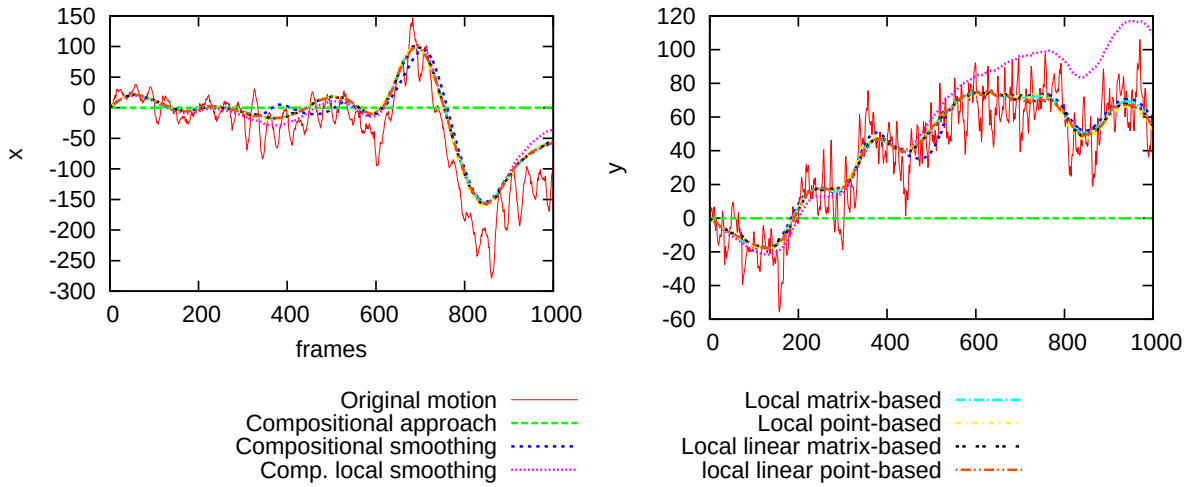

Figure 9. *Walking* compositional trajectories *of the central pixel obtained by applying successively all transition homographies on the central point in the first frame. The left and right graphics show, respectively, the x and y components. All smoothing strategies are displayed.*

The linear scale-space analysis performs a multiscale frequency analysis of the *Walking* sequence. The peaks in the third and fourth DFT graphics around one Hertz can be interpreted to correspond to the horizontal stepwise walking of the person, while the smaller peak at two Hertz in the fourth graphic would correspond to the vertical stepwise period. We found this frequency pattern in all walking sequences we tested.

Similar to the previous graphics, we show, in Figure 11, the linear scale-space of the zoom parameter. The smoothed upper curve is above the straight blue line which corresponds to zoom $= 1$. Thus this smoothed curve gives an excellent account of the walking speed, which increases toward the end of the walk. On the right, the DFT peaks at approximately 1Hz and at 2Hz retrieves the periodicity of this regular movement, but the spectrum also shows

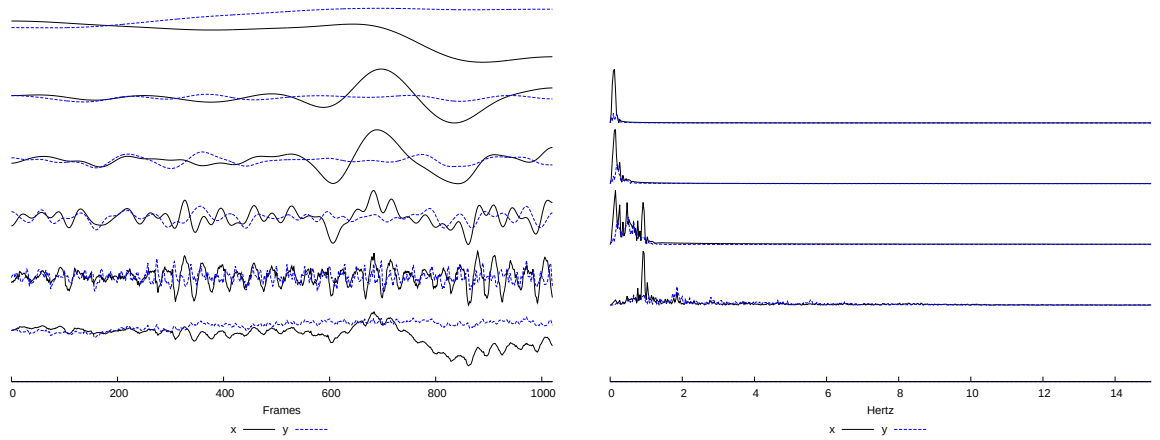

Figure 10. *Walking sequence. Left, from bottom to top: Virtual trajectory of the central point using the original transformations (Definition 6.1); difference between the smoothed trajectory with* $\sigma := 10$ *and the original trajectory; difference between the solution with* $\sigma := 20$ *and* $\sigma := 10$*; difference between the solution with* $\sigma := 40$ *and* $\sigma := 20$*; difference between the solution with* $\sigma := 80$ *and* $\sigma := 40$*; finally the original signal smoothed with* $\sigma := 80$*. Right: DFT signals of the corresponding graphics on the left. Each graphic depicts the* x *and* y *components.*

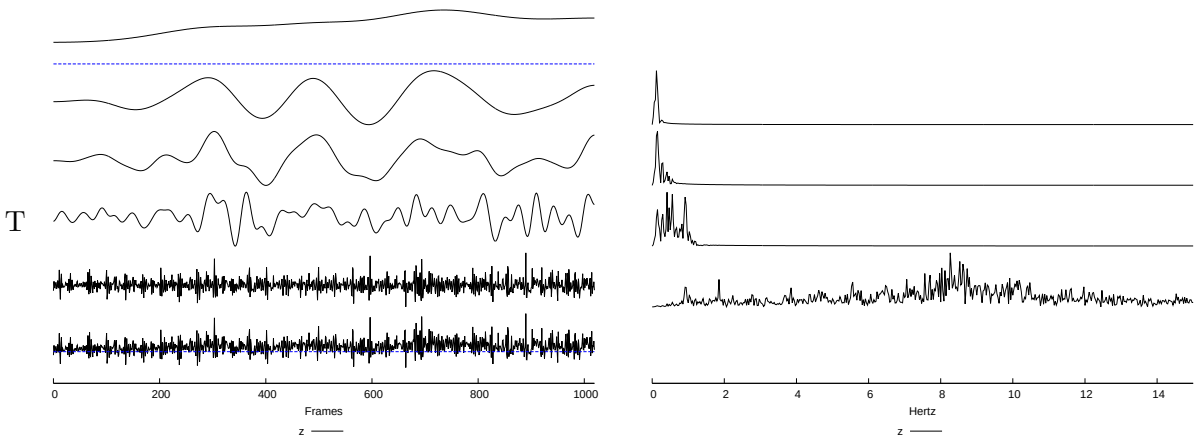

Figure 11. *Walking linear scale-space analysis (zoom).*

high frequencies at 8 Hz, which may be due to a random bouncing of the hanging camera. Figure 12 shows a similar result for the rotation. We shall observe a more clear-cut frequency pattern on running ego-motions when the camera is fixed to the body.

In the next experiment, *Earthquake*, we used a sequence taken from a surveillance camera recording an earthquake. This sequence is composed of 5.342 frames and is more challenging, because there are rapid and strong vibrations in some parts of the video.

Figure 13 shows linear scale-spaces for the trajectory and the rotation of the camera. Interestingly, the central point moves to the right and top in the sequence. No dominant frequency is observable. The smoothed rotation indicates that the pole rotates and rotates

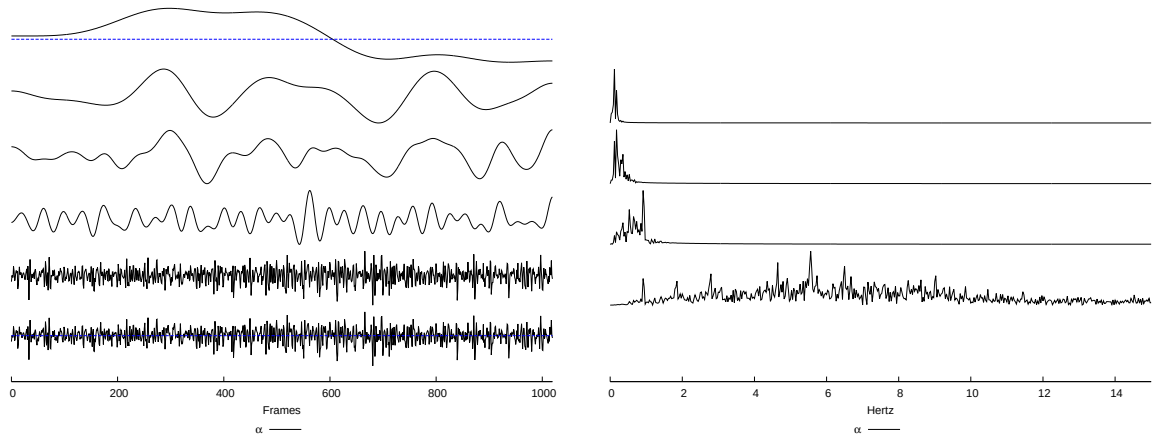

Figure 12. *Walking linear scale-space analysis (rotation).*

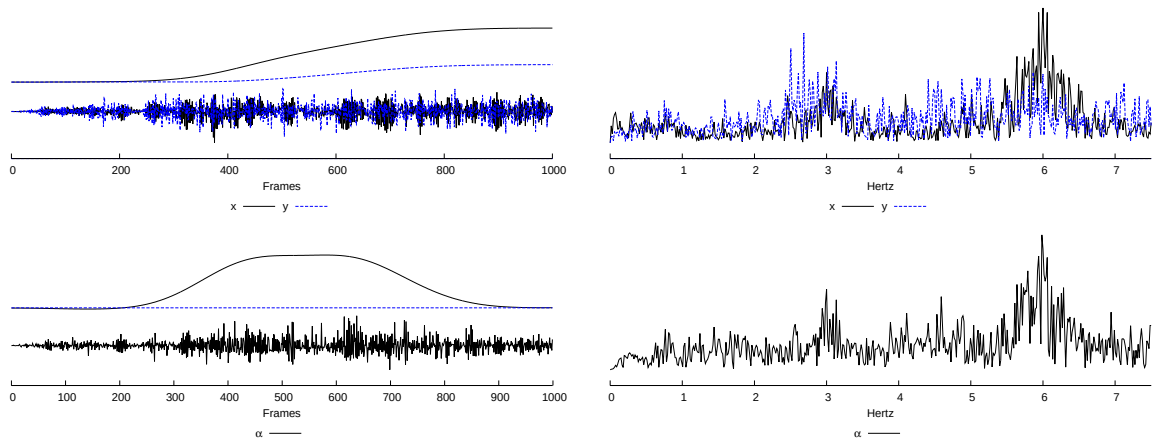

Figure 13. *Earthquake linear scale-space analysis: Top, scale-space of the central point and its DFT analysis; bottom, scale-space of the rotation signal. In each row, the top curve displays the original signal smoothed with* $\sigma := 80$ *and the bottom curve is the difference between the smoothed signal with* $\sigma := 10$ *and the original one. The right curve is its DFT.*

back during the earthquake.

In the next experiment, we analyze the sequence of a man running with a camera fixed on his head. We observe in Figure 14 the characteristic motion of a running person, where the displacement is periodic in both the horizontal and vertical directions. There is a dominant peak in the x component around 1,5 Hz, which denotes the oscillation from left to right. In the y component, there is a peak at 3 Hz corresponding to the vertical oscillation.

In the zoom figure, we observe that the graphics present fast periodic motions, which are, on average, above the blue line, corresponding to a static zoom. There is a steady zoom-in due to the forward motion of the camera. On the other hand, the oscillations are probably due to the balancing of the head while running.

In the last experiment, we analyze the video of a man running with a camera on his chest.

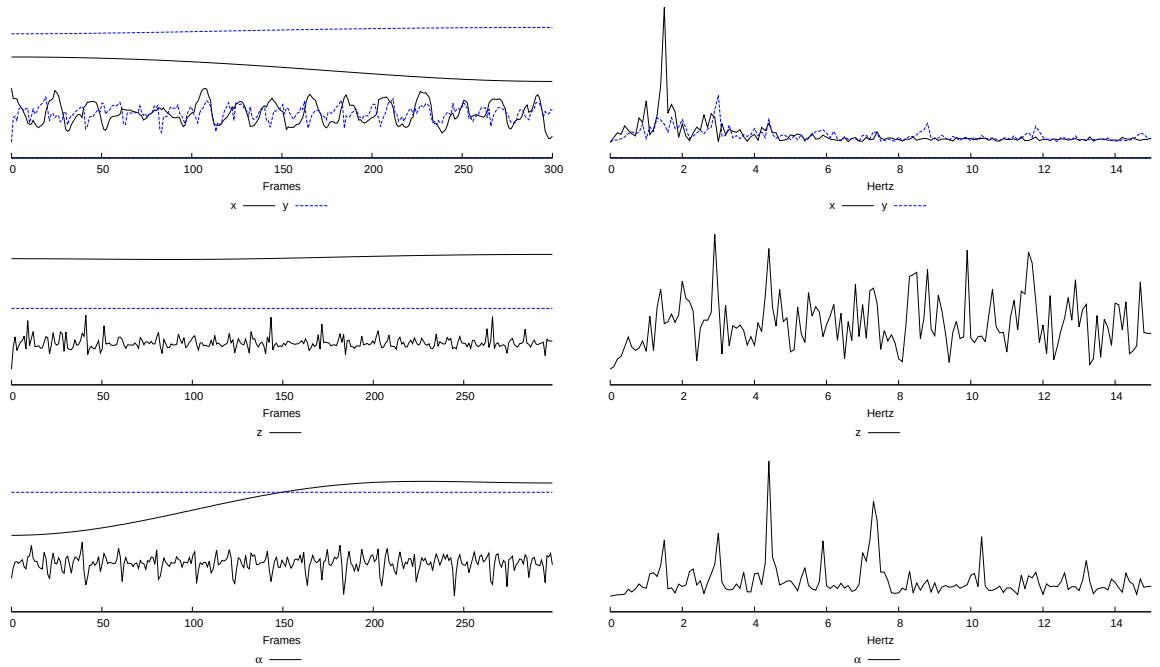

Figure 14. *Man running with a GoPro on his head, linear scale-space analysis. In the top left figure, we show the scale-space corresponding to the trajectory of the central point; in the middle left figure, we show the scale-space of the zoom signal; and, in the bottom left figure, we show the scale-space of the rotation signal. In these figures, the graphics above show the original signal smoothed with* $\sigma := 80$ *and the graphics below show the difference between the smoothed signal with* $\sigma := 10$ *and the original one. The right figures depict the DFT signals of these graphics.*

The results are similar to the previous experiment, although the shaking is more important than when the camera is fixed to the head. This is reasonable since persons tend to stabilize their head when they run or walk.

The trajectory, in the linear scale-space graphics of Figure 15, seems to be more regular and periodic; see the two graphics on the bottom. This is due to the fact that the chest is more rigid and better represents the motion of the run. The peaks are also situated about 1,5Hz for the horizontal displacement and slightly above 3Hz for the vertical one.

The zoom and rotation linear scale-spaces are also similar to the previous experiment, although the rotation seems to be more regular and aligned with the horizontal displacement. This may be produced by the motion from side to side of the chest while moving the arms, which is synchronized with the motion of the legs.

7. Conclusion.

We proposed a classification of motion compensation strategies for video stabilization and characterized the different types of boundary conditions for the involved smoothing.

We classified the smoothing strategies in compositional and additive approaches. The former are further divided into global methods, which compose the transformations from a given reference frame, and local methods, which compose the transformations in a neighborhood

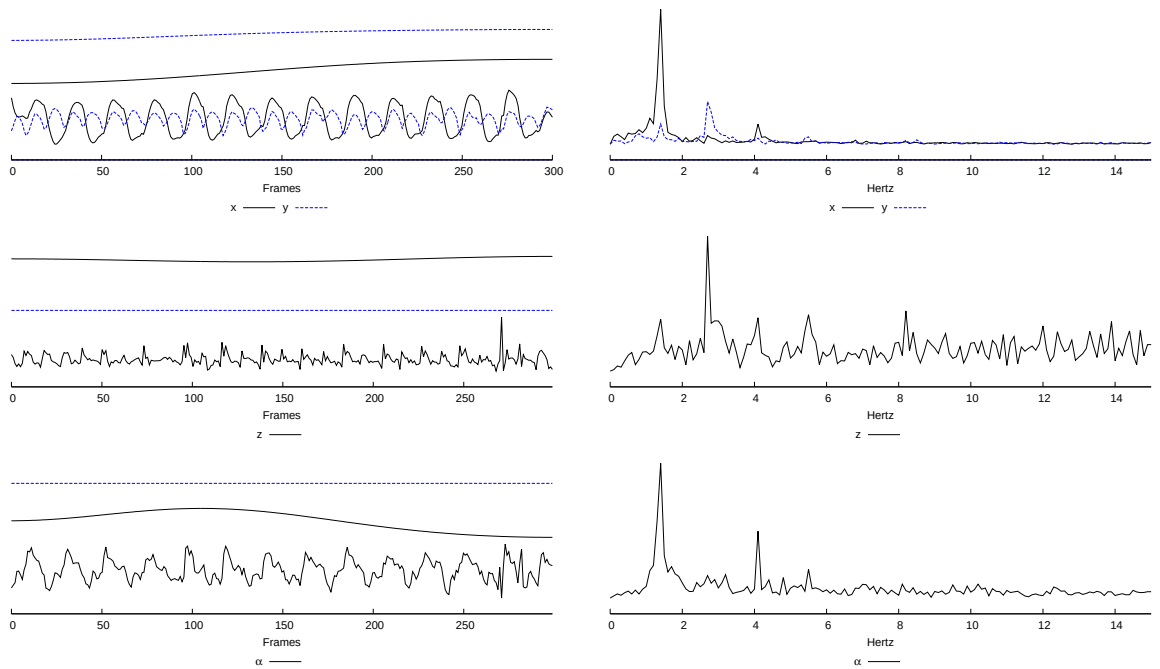

Figure 15. *Man running with a GoPro on his chest, linear scale-space analysis. In the top left figure, we show the scale-space corresponding to the trajectory of the central point; in the middle left figure, we show the scale-space of the zoom signal; and, in the bottom left figure, we show the scale-space of the rotation signal. In these figures, the graphics above show the original signal smoothed with $\sigma := 80$ and the graphics below show the difference between the smoothed signal with $\sigma := 10$ and the original one. The right figures depict the DFT signals of these graphics.*

around each image.

We added to the list two linear methods working on the coefficients of the transformations or on virtual trajectories obtained as integrals of the instant velocity. The experiments showed that these methods provide the best results in terms of preservation of the video content after stabilization.

We also found that local smoothing approaches are more flexible and robust to the errors introduced in the motion estimation step. The accumulation errors are only restricted to the size of the convolution kernel. These are more suitable for affinities and homographies.

Analyzing the frequency patterns and the smooth tendency in the scale space of the virtual trajectories yields crucial temporal information such as frequency peaks linked to periodic ego-motions and a smoothed evaluation of the movement forward and of the rotations. This scale-space analysis was applied to extremely shaky video to check that such information can be reliably extracted.

REFERENCES

[1] W. G. Aguilar and C. Angulo, *Real-time model-based video stabilization for microaerial vehicles*, Neural Process. Lett., 43 (2016), pp. 459–477.

[2] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel, *Axioms and fundamental equations of image processing*, Arch. Ration. Mech. Anal., 123 (1993), pp. 199–257.
[3] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, *A variational framework for exemplar-based image inpainting*, Internat. J. Comput. Vis., 93 (2011), pp. 319–347.
[4] S. Baker, R. Gross, T. Ishikawa, and I. Matthews, *Lucas-Kanade 20 Years On: A Unifying Framework: Part 2*, Tech. report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2003.
[5] S. Baker and I. Matthews, *Lucas-Kanade 20 years on: A unifying framework*, Internat. J. Comput. Vis., 56 (2004), pp. 221–255.
[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, *Speeded-up robust features (surf)*, Comput. Vis. Image Understanding, 110 (2008), pp. 346–359.
[7] S. Bell, A. Troccoli, and K. Pulli, *A non-linear filter for gyroscope-based video stabilization*, in European Conference on Computer Vision, Springer, New York, 2014, pp. 294–308.
[8] P. Bouthemy, M. Gelgon, and F. Ganansia, *A unified approach to shot change detection and camera motion characterization*, IEEE Trans. Circuits Syst. Video Tech., 9 (1999), pp. 1030–1044.
[9] M. Brown and D. G. Lowe, *Recognising panoramas*, in ICCV, Vol. 3, IEEE, Washington, DC, 2003, 1218.
[10] C. Buehler, M. Bosse, and L. McMillan, *Non-metric image-based rendering for video stabilization*, in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Vol. 2, IEEE, Washington, DC, 2001, pp. II–609–II–614.
[11] B.-Y. Chen, K.-Y. Lee, W.-T. Huang, and J.-S. Lin, *Capturing intention-based full-frame video stabilization*, Comput. Graphics Forum, 27 (2008), pp. 1805–1814.
[12] A. Criminisi, P. Pérez, and K. Toyama, *Region filling and object removal by exemplar-based image inpainting*, IEEE Trans. Image Process., 13 (2004), pp. 1200–1212.
[13] K. Daniels, V. Milenkovic, and D. Roth, *Finding the largest area axis-parallel rectangle in a polygon*, Comput. Geom., 7 (1997), pp. 125–148.
[14] J. Delon, *Movie and video scale-time equalization application to flicker reduction*, IEEE Trans. Image Process., 15 (2006), pp. 241–248.
[15] P.-E. Forssén and E. Ringaby, *Rectifying rolling shutter video from hand-held devices*, in 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2010, pp. 507–514.
[16] P. Getreuer, *A survey of Gaussian convolution algorithms*, Image Processing On Line, 3 (2013), pp. 286–310, https://doi.org/10.5201/ipol.2013.87.
[17] M. L. Gleicher and F. Liu, *Re-cinematography: Improving the camerawork of casual video*, ACM Trans. Multimedia Comput. Commun. Appl., 5 (2008), 2.
[18] A. Goldstein and R. Fattal, *Video stabilization using epipolar geometry*, ACM Trans. Graphics, 31 (2012), 126.
[19] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, *Calibration-free rolling shutter removal*, in 2012 IEEE International Conference on Computational Photography (ICCP), IEEE, Washington, DC, 2012, pp. 1–8.
[20] M. Grundmann, V. Kwatra, and I. Essa, *Auto-directed video stabilization with robust L1 optimal camera paths*, in 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2011, pp. 225–232.
[21] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, *Real-time scene stabilization and mosaic construction*, in Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision, IEEE, Washington, DC, 1994, pp. 54–62.
[22] C. Harris and M. Stephens, *A combined corner and edge detector*, in Alvey Vision Conference, Vol. 15, Manchester, UK, Citeseer, 1988, 50.
[23] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, Cambridge, UK, 2004.
[24] J. Hedborg, P. E. Forssén, M. Felsberg, and E. Ringaby, *Rolling shutter bundle adjustment*, in 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Washington, DC, 2012, pp. 1434–1441.
[25] T. Huynh, M. Fritz, and B. Schiele, *Discovery of activity patterns using topic models*, in Proceedings of the 10th International Conference on Ubiquitous Computing, ACM, New York, 2008, pp. 10–19.

[26] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy, *Digital Video Stabilization and Rolling Shutter Correction Using Gyroscopes*, CSTR 2011-03, Stanford University, Stanford, CA, 2011.
[27] J.-G. Kim, H. S. Chang, J. Kim, and H.-M. Kim, *Efficient camera motion characterization for mpeg video indexing*, in IEEE International Conference on Multimedia and Expo (ICME), Vol. 2, IEEE, Washington, DC, 2000, pp. 1171–1174.
[28] S.-J. Ko, S.-H. Lee, and K.-H. Lee, *Digital image stabilizing algorithms based on bit-plane matching*, IEEE Trans. Consumer Electronics, 44 (1998), pp. 617–622.
[29] J. J. Koenderink, *The structure of images*, Biolog. Cybernet., 50 (1984), pp. 363–370.
[30] J. Kopf, 360^o *video stabilization*, ACM Trans. Graphics, 35 (2016), 195.
[31] J. Kopf, M. F. Cohen, and R. Szeliski, *First-person hyper-lapse videos*, ACM Trans. Graphics, 33 (2014), 78.
[32] D. Kundur and D. Hatzinakos, *Blind image deconvolution*, IEEE Signal Process. Mag., 13 (1996), pp. 43–64.
[33] I. Laptev, *On space-time interest points*, Internat. J. Comput. Vis., 64 (2005), pp. 107–123.
[34] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung, *Video stabilization using robust feature trajectories*, in Proceedings of the 12th IEEE International Conference on Computer Vision, IEEE, Washington, DC, 2009, pp. 1397–1404.
[35] T. Lindeberg, *Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention*, Internat. J. Comput. Vis., 11 (1993), pp. 283–318.
[36] T. Lindeberg, *Scale-space theory: A basic tool for analyzing structures at different scales*, J. Appl. Statist., 21 (1994), pp. 225–270.
[37] T. Lindeberg, *On the axiomatic foundations of linear scale-space*, in Gaussian Scale-Space Theory, Springer, New York, 1997, pp. 75–97.
[38] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Springer Internat. Ser. Engrg. Comput. Sci. 256, Springer Science & Business Media, Dordrecht, The Netherlands, 2013.
[39] A. Litvin, J. Konrad, and W. C. Karl, *Probabilistic video stabilization using Kalman filtering and mosaicing*, in Electronic Imaging 2003, International Society for Optics and Photonics (SPIE), Bellingham, WA, 2003, pp. 663–674.
[40] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, *Content-preserving warps for 3D video stabilization*, ACM Trans. Graphics, 28 (2009), 44.
[41] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, *Subspace video stabilization*, ACM Trans. Graphics, 30 (2011), 4.
[42] S. Liu, M. Li, S. Zhu, and B. Zeng, *Codingflow: Enable video coding for video stabilization*, IEEE Trans. Image Process., 26 (2017), pp. 3291–3302.
[43] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng, *MeshFlow: Minimum latency online video stabilization*, in Computer Vision—ECCV 2016, Springer International Publishing, Cham, 2016, pp. 800–815.
[44] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun, *Video stabilization with a depth camera*, in 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2012, pp. 89–95.
[45] S. Liu, L. Yuan, P. Tan, and J. Sun, *Bundled camera paths for video stabilization*, ACM Trans. Graphics, 32 (2013), 78.
[46] S. Liu, L. Yuan, P. Tan, and J. Sun, *Steadyflow: Spatially smooth optical flow for video stabilization*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2014, pp. 4209–4216.
[47] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, Internat. J. Comput. Vis., 60 (2004), pp. 91–110.
[48] B. D. Lucas, T. Kanade, et al., *An iterative image registration technique with an application to stereo vision*, in IJCAI, Vol. 81, Vancouver, Canada, 1981, pp. 674–679.
[49] J. Matas, O. Chum, M. Urban, and T. Pajdla, *Robust wide-baseline stereo from maximally stable extremal regions*, Image Vis. Comput., 22 (2004), pp. 761–767.
[50] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, *Full-frame video stabilization with motion inpainting*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 1150–1163.

[51] 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, 2 (2012), pp. 56–73, https://doi.org/10.5201/ipol.2012.mmm-oh.

[52] C. Morimoto and R. Chellappa, *Fast electronic digital image stabilization*, in Proceedings of the 13th International Conference on Pattern Recognition, Vol. 3, IEEE, Washington, DC, 1996, pp. 284–288.

[53] C. Morimoto and R. Chellappa, *Evaluation of image stabilization algorithms*, in Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 5, IEEE, Washington, DC, 1998, pp. 2789–2792.

[54] M. Okade and P. K. Biswas, *Video stabilization using maximally stable extremal region features*, Multimedia Tools Appl., 68 (2014), pp. 947–968.

[55] H. Pirsiavash and D. Ramanan, *Detecting activities of daily living in first-person camera views*, in 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2012, pp. 2847–2854.

[56] E. Ringaby and P.-E. Forssén, *Efficient video rectification and stabilisation for cell-phones*, Internat. J. Comput. Vis., 96 (2012), pp. 335–352.

[57] J. Sánchez, *The inverse compositional algorithm for parametric registration*, Image Processing On Line, 6 (2016), pp. 212–232, https://doi.org/10.5201/ipol.2016.153.

[58] J. Shi and C. Tomasi, *Good features to track*, in 1994 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Washington, DC, 1994, pp. 593–600.

[59] B. M. Smith, L. Zhang, H. Jin, and A. Agarwala, *Light field video stabilization*, in 12th IEEE International Conference on Computer Vision, IEEE, Washington, DC, 2009, pp. 341–348.

[60] J. Sporring, M. Nielsen, L. Florack, and P. Johansen, *Gaussian Scale-Space Theory*, Comput. Imaging Vis. 8, Springer Science & Business Media, Dordrecht, The Netherlands, 2013.

[61] E. H. Spriggs, F. De La Torre, and M. Hebert, *Temporal segmentation and activity classification from first-person sensing*, in IEEE Conference on Computer Vision and Pattern Recognition Workshops, IEEE, Washington, DC, 2009, pp. 17–24.

[62] R. Szeliski, *Computer Vision Algorithms and Applications*, Springer, London, New York, 2011.

[63] R. Szeliski and H.-Y. Shum, *Creating full view panoramic image mosaics and environment maps*, in Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, ACM, New York, Addison–Wesley, Reading, MA, 1997, pp. 251–258.

[64] B. ter Haar Romeny and B. M. ter Haar Romeny, eds., *Scale-Space Theory in Computer Vision: First International Conference, Scale-Space'97 (Utrecht, The Netherlands)*, Lecture Notes in Comput. Sci. 1252, Springer-Verlag, Berlin, 1997.

[65] M. Trajković and M. Hedley, *Fast corner detection*, Image Vis. Comput., 16 (1998), pp. 75–87.

[66] F. Vella, A. Castorina, M. Mancuso, and G. Messina, *Digital image stabilization by adaptive block motion vectors filtering*, IEEE Trans. Consumer Electronics, 48 (2002), pp. 796–801.

[67] Y.-S. Wang, F. Liu, P.-S. Hsu, and T.-Y. Lee, *Spatially and temporally optimized video stabilization*, IEEE Trans. Visualization Comput. Graphics, 19 (2013), pp. 1354–1361.

[68] J. Weickert, S. Ishikawa, and A. Imiya, *On the history of Gaussian scale-space axiomatics*, in Gaussian Scale-Space Theory, Springer, New York, 1997, pp. 45–59.

[69] Y. Wexler, E. Shechtman, and M. Irani, *Space-time completion of video*, IEEE Trans. Pattern Anal. Mach. Intell., 29 (2007), pp. 463–476.

[70] A. Witkin, *Scale-space filtering: A new approach to multi-scale description*, in IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'84, Vol. 9, IEEE, Washington, DC, 1984, pp. 150–153.