Jitter-free registration for Unmanned Aerial Vehicle Videos

Pierre Lemaire, Carlos F Crispim-Junior, Lionel Robinault, Laure Tougne

Jitter-free registration for Unmanned Aerial Vehicle Videos

Pierre Lemaire[1], Carlos Fernando Crispim Junior[1], Lionel Robinault[2], and Laure Tougne[1]

[1] Université Lyon 2, LIRIS, UMR5205, F-69676, France
pierre.lemaire@liris.cnrs.fr carlos.crispim-junior@liris.cnrs.fr
laure.tougne@liris.cnrs.fr
[2] Foxstream, Vaulx-en-Velin, F-69120, France l.robinault@foxstream.fr

Abstract. Unmanned Aerial Vehicles (UAVs), such as tethered drones, become increasingly popular for video acquisition, within video surveillance or remote, scientific measurement contexts. However, UAV recordings often present an unstable, variable viewpoint that is detrimental to the automatic exploitation of their content. This is often countered by one amongst two strategies, video registration and video stabilization, which are usually affected by distinct issues, namely jitter and drifting. This paper proposes a hybrid solution between both techniques that produces a jitter-free registration. A lightweight implementation enables real time, automatic generation of videos with a constant viewpoint from unstable video sequences acquired with stationary UAVs. Performance evaluation is carried out using video recordings from traffic surveillance scenes up to 15 minutes long, including multiple mobile objects.

Keywords: Motion Estimation, Video Registration, Video Stabilization, Unmanned Aerial Vehicles

1 Introduction

Unmanned Aerial Vehicles (UAVs), are becoming increasingly popular for tasks such as video surveillance or remote data acquisition [15]. Tethered drones [10] can now fly during several hours up to 50 meters above ground in stationary flight. Their video flux looks a lot like that of a classic surveillance camera. However, their lack of stability makes tasks such as background subtraction or objects tracking more complex than with a fixed viewpoint (Fig. 1.a and 1.b). This paper proposes a real-time, online method to convert videos acquired from a stationary drone into jitter-free, single viewpoint videos as much as possible.

Real-world applications for stationary UAVs such as traffic monitoring or crowd surveillance often present a high density of mobile objects, which may provoke drifting and jitter issues over time. Yet, prior works on UAVs image stabilization techniques have been studied on datasets that include very few mobile objects and rather short sequences [13,2].

We propose to tackle the problem of lengthy sequences with multiple mobile objects. In order to leave room for further analysis processes, our solution needs

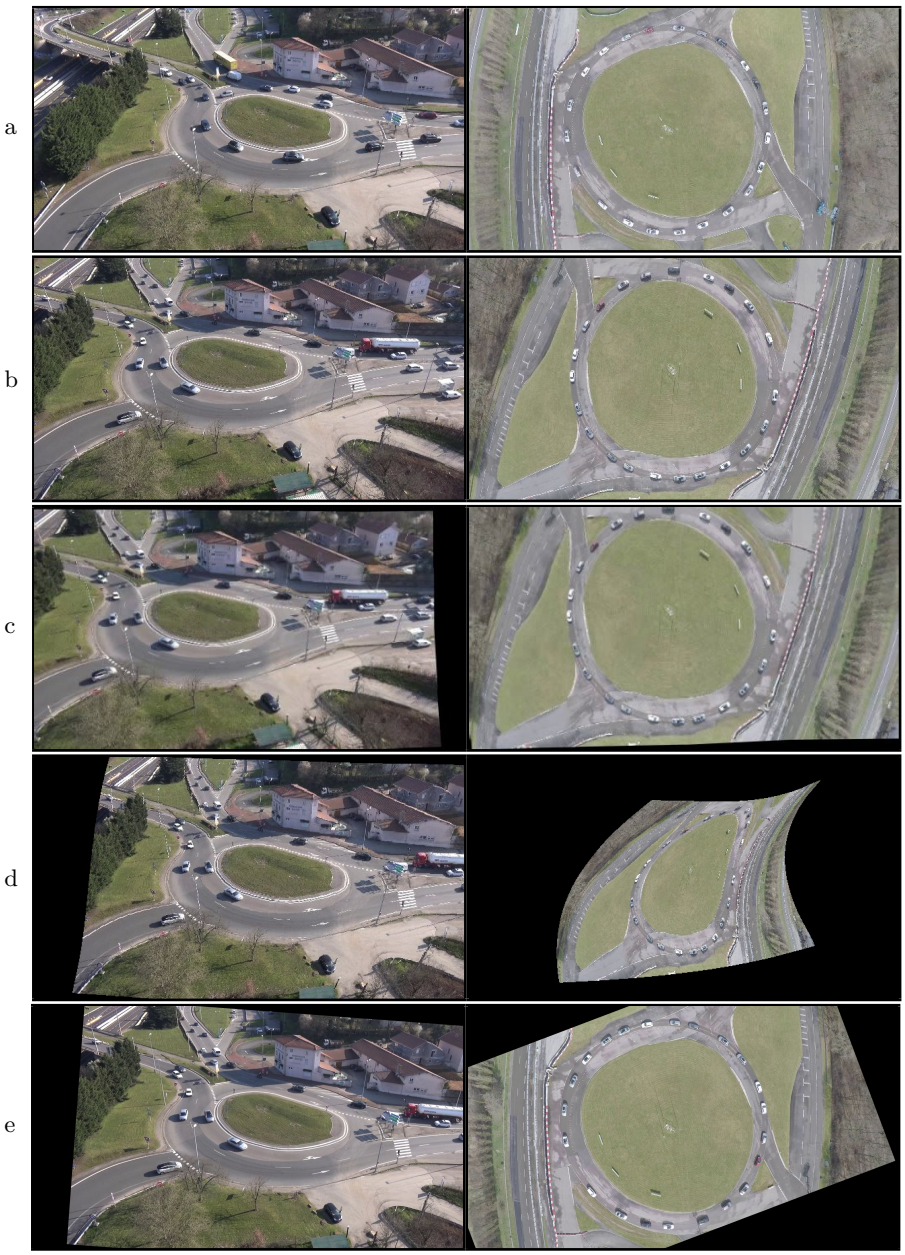

Fig. 1. Images extracted from the M4 (left column) and the C2 (right column) sequences in our database. a: first frame, used as a reference image. b: current frame, after approx. 15 seconds (resp. 45 seconds) on left M4 (resp. C2), which we want to register to the first frame of the sequence. c: the output of StabNet [16]. d: the output of CNN-Registration [18]. e: the output of the proposed method.

to be online and computationally low-cost. For this purpose, we propose a generic model which can be applied to 2D rigid motion estimation methods. We show how to combine stabilization and registration techniques, and we apply this method to a lightweight 2D-rigid transformation registration algorithm.

2 Prior work

Producing a constant viewpoint video from a mobile camera is quite equivalent to determining the camera orientation. Determining the extrinsic parameters of a monocular camera within a 3D environment is a problem typically studied by Structure from Motion (SfM) [12] or Simultaneous Localization and Mapping (SLAM) approaches, some of which can operate in real time [11]. However, the latter are mostly designed to work on static environments and rely on parallax, *ie.* when there is enough camera movements to infer the 3D structure of the scene.

Video registration and video stabilization tackle this problem by searching for an image transformation that optimally compensates for camera motion. In the first case, this transformation is estimated between the current frame and a reference image. In the stabilization case, we calculate a trajectory, defined as a combination of consecutive inter-frame motion estimations. This trajectory is then filtered and the image is reprojected so to follow the desired, smooth trajectory.

In the UAV context, authors have stated that the direct application of a registration method to a reference frame leads to unsatisfying, jittery results [13,2,1]. Jitter is often linked to an unstable image source (handheld camera, mechanical high frequency noise, *etc.*). However, it can also correspond to a high-frequency noise caused by the image motion compensation itself. To our understanding, this happens because most registration methods are based on a sparse feature points matching solution, generally accompanied by a selection of inliers and outliers technique such as RANSAC. The intermittent presence of points caused by thresholds in the matching or the inliers selection processes may cause such high-frequency noise.

Classic video stabilization methods such as [8] and [6] have been adapted to the UAV context, sometimes associated with video stitching [7]. The motion estimation is often performed with a very popular approach known as Kanade Lukas Tracker (KLT) [14] but other techniques have been proposed, such as a specific optical flow model which enforces spatial coherence [9]. Most methods are able to handle mobile camera, and thus do not assume the existence of a constant background, which however applies in our context. They eliminate jitter very well, but they tend to imply drifting, which is the tendency to slowly change viewpoint over time.

More recently, convolutional neural networks have been applied to both registration [18] and stabilization [16] domains. Both methods have proposed to use a rich warping model based on Thin Plate Splines (TPS). While such approaches

look promising, their direct application on our data proved problematic. We may observe on Figs. 1.c (both columns) and 1.d (right column) that 3 out of 4 frames are misaligned relatively to the reference image (Fig. 1.a). Authors of StabNet [16] based their approach on a siamese convolutional network that was trained thanks to a stabilization database. This database was acquired with the help of a single handling device to which 2 different cameras were attached, only one of which was physically stabilized with a gimbal. Such ground-truth is not available in our settings. Moreover, this method still does not assume the presence of a constant background to which it should register. CNN-Registration [18] is able to handle large appearance changes and seems suitable to handle long-term registration with important lighting variations and the presence of multiple mobile objects. However, our experiments have shown that this approach is not invariant to rotation (e.g., it is not able to handle a video rotated to some extent, Fig. 1.d) or to very large displacement. In our context, applying it would thus require some prior alignment step, which confirms the need for a simple and robust registration method that takes temporal data into account.

3 Modelling the problem

The idea behind a video stabilization or registration algorithm is to compensate for undesired camera motion, while preserving the image content variability over time. At first, we need to define the degrees of freedom of our problem.

In first approximation, the effects of camera movements can be modeled and compensated through 2D-rigid warping transforms. On stationary drones, the camera is mostly affected by undesired, relatively low magnitude Yaw, Pitch and Roll motion (following the Tait Bryan chained rotations convention). Since the drone is never perfectly stationary, additional undesired 3D translational motion of the camera in space as well as the 3D geometry of the scene add up to the complexity of our problem.

Eventually, we estimate the camera motion between two images Im_i and Im_j through a 2D linear transformation matrix $\tilde{\mathcal{M}}(i,j)$. It is defined by an unique quartet of parameters (t_x, t_y, α, s) (resp. translation along the horizontal and vertical axis, rotation of angle α, and a positive scale in the 2D plane) which are used to approximate the effects on the image of a physical 3D motion performed by the drone (resp. Yaw, Pitch, Roll and translation along the optical axis). Warping Im_i according to the transformation matrix $\tilde{\mathcal{M}}(i,j)$ aims at setting it in the closest possible viewpoint to Im_j. Conversely, warping Im_j according to $\tilde{\mathcal{M}}(j,i) = \tilde{\mathcal{M}}(i,j)^{-1}$ sets it to the closest possible viewpoint to Im_i.

In any case, we rely on the estimation of the motion between two images Im_i and Im_j, for which we propose the following decomposition:

$$\tilde{\mathcal{M}}(i,j) = \mathcal{E}_{\tilde{\mathcal{M}}}(i,j) \times \mathcal{M}_{cam}(i,j) \tag{1}$$

where:

- $\tilde{\mathcal{M}}(i,j)$ is the estimated camera motion between Im_i and Im_j

- $\mathcal{M}_{cam}(i,j)$ corresponds to the motion associated to the actual, physical camera movement, measured as background motion between Im_i and Im_j
- $\mathcal{E}_{\tilde{\mathcal{M}}}(i,j)$ corresponds to a motion estimation error.
  $\mathcal{E}_{\tilde{\mathcal{M}}}(i,j)$, $\tilde{\mathcal{M}}(i,j)$ and $\mathcal{M}_{cam}(i,j)$ are all expressed as linear transformation matrices.

A lot of motion estimation or registration methods are available in the literature, ranging from holistic [5] to sparse [14], with various properties and advantages. Equation (1) can be used to characterize any movement estimation algorithm that outputs a 2D linear transform.

Registering a video is the problem of canceling the term $\mathcal{M}_{cam}$ over the course of a video. With a reference frame denoted as 0, the applied warping can be expressed as

$$\mathcal{W}(i) = \tilde{\mathcal{M}}(0,i)^{-1} = \mathcal{M}_{cam}(0,i)^{-1} \times \mathcal{E}_{\tilde{\mathcal{M}}}(0,i)^{-1} \tag{2}$$

Stabilizing a video is the problem of smoothing $\mathcal{M}_{cam}$ over the course of a video. This is performed by constructing a trajectory, which is defined as

$$\begin{cases} \mathcal{T}_{\tilde{\mathcal{M}}}(0) = \mathcal{I} \\ \mathcal{T}_{\tilde{\mathcal{M}}}(i) = \tilde{\mathcal{M}}(i-1,i) \times \mathcal{T}_{\tilde{\mathcal{M}}}(i-1) \text{ with } i > 0 \end{cases} \tag{3}$$

which we denote as:

$$\mathcal{T}_{\tilde{\mathcal{M}}}(i) = \prod_{k=1}^{\overset{\curvearrowleft}{i}} \tilde{\mathcal{M}}(k-1,k) \tag{4}$$

The left arrow sign ($\curvearrowleft$) means that we perform a left-hand product.

Then, we filter $\mathcal{T}_{\tilde{\mathcal{M}}}$ over time: the warping applied to the original images can be seen as the difference between the filtered trajectory and the original trajectory.

$$\mathcal{W}(i) = \mathcal{F}\big(\mathcal{T}_{\tilde{\mathcal{M}}}\big)(i) \times \mathcal{T}_{\tilde{\mathcal{M}}}(i)^{-1} \tag{5}$$

where $\mathcal{F}\big(\mathcal{T}_{\tilde{\mathcal{M}}}\big)(i)$ is the output at frame i of a smoothing filter applied on the set of trajectories $\mathcal{T}_{\tilde{\mathcal{M}}}$. It is also expressed as a 2D linear transform matrix. Any output of $\mathcal{F}$ should be a plausible approximation given the physical constraints of the problem. In practice, one can filter independently t_x, t_y, α and s. For real-time, online application, a Kalman Filter [17] can be used.

Given Eq. (1), we can reformulate Eq. (3) as follows:

$$\mathcal{T}_{\tilde{\mathcal{M}}}(i) = \prod_{k=1}^{\overset{\curvearrowleft}{i}} \big(\mathcal{E}_{\tilde{\mathcal{M}}}(k-1,k) \times \mathcal{M}_{cam}(k-1,k)\big) \tag{6}$$

By definition,

$$\mathcal{M}_{cam}(0,i) = \prod_{k=1}^{\overset{\curvearrowleft}{i}} \mathcal{M}_{cam}(k-1,k) \tag{7}$$

In the general case, we cannot develop any further Eq. (6). However, we can introduce an equivalent error term such as Eq. (3) becomes:

$$\mathcal{T}_{\tilde{\mathcal{M}}}(i) = \mathcal{E}^{equiv}_{\tilde{\mathcal{M}}}(0,i) \times \mathcal{M}_{cam}(0,i) \tag{8}$$

The more dissimilar Im_i and Im_j, the more significant $\mathcal{E}_{\tilde{\mathcal{M}}}(i,j)$ is likely to be. Consecutive images being rather similar, they usually yield a $\mathcal{E}_{\tilde{\mathcal{M}}}$ term of low magnitude. However, in such cases, foreground objects often perform little movement from Im_i to Im_j. When i and j are close in time, a part of the term $\mathcal{E}_{\tilde{\mathcal{M}}}(i,j)$ may correspond to light foreground motion that was wrongly considered as background motion by the motion estimator. Such error accumulates into Eq. (6) to form a drifting trajectory (Eq. (8)). This drifting error term explains why it is not recommended to simply use $\mathcal{T}^{-1}_{\tilde{\mathcal{M}}}$ as a registration solution.

Most of the literature in stabilization and registration topics is focused on minimizing the term $\mathcal{E}_{\tilde{\mathcal{M}}}$ within the motion estimation step. This minimization is essential towards achieving good performance, but the existence of such error is unavoidable. However, its nature tends to vary from jitter in a registration case to drifting in a stabilization case. We show how to take advantage of both jittery and drifting behaviors to propose an efficient and low-cost solution towards jitter-free constant viewpoint generation.

4 Proposed method

In this section, we show how to efficiently combine registration and stabilization approaches into a single hybrid method (Fig. 2). From now on, we will denote by $\tilde{\mathcal{M}}^s$ (resp. $\tilde{\mathcal{M}}^r$ the specific motion estimator for the stabilization (resp. registration) part of the proposed method.

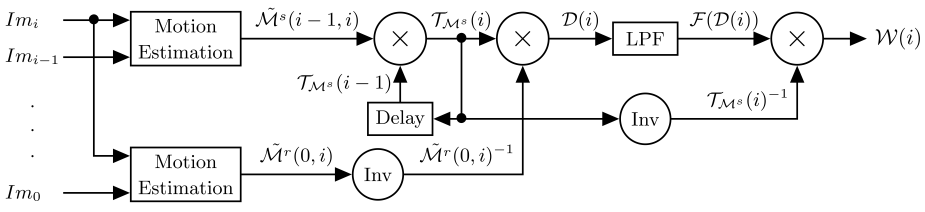

Fig. 2. Overview of the proposed method. Inv corresponds to a matrix inversion; LPF stands for Low-Pass Filter.

The idea is to calculate the product between the trajectory $\mathcal{T}_{\tilde{\mathcal{M}}^s}$ of a stabilization method, and the correction $\tilde{\mathcal{M}}^r(0,i)^{-1}$ applied in a registration method.

$$\mathcal{D}(i) = \mathcal{T}_{\tilde{\mathcal{M}}^s}(i) \times \tilde{\mathcal{M}}^r(0,i)^{-1} \tag{9}$$

Following the model proposed in Eq. (1), as reported in Eqs. (2) and (8), we can reformulate Eq. (9) as:

$$\mathcal{D}(i) = \mathcal{E}^{equiv}_{\tilde{\mathcal{M}}^s}(0,i) \times \left(\mathcal{E}_{\tilde{\mathcal{M}}^r}(0,i)\right)^{-1} \tag{10}$$

As suggested previously, this matrix is essentially a product of a smooth, low frequency drifting error ($\mathcal{E}^{equiv}_{\tilde{\mathcal{M}}^s}(0,i)$) and a jittery, high frequency error ($\mathcal{E}_{\tilde{\mathcal{M}}^r}(0,i)$). Filtering $\mathcal{D}$ allows us to isolate the drifting component.

$$\mathcal{F}\left(\mathcal{D}(i)\right) \approx \mathcal{E}^{equiv}_{\tilde{\mathcal{M}}^s}(0,i) \tag{11}$$

Combining the output of this filter to $\mathcal{T}^{-1}_{\tilde{\mathcal{M}}^s}$ finally allows us to obtain a jitter-free video registration on long sequences, without the need for particularly elaborate motion estimation techniques. Finally, the applied correction is the following:

$$\mathcal{W}(i) = \mathcal{F}\left(\mathcal{D}(i)\right) \times \mathcal{T}_{\tilde{\mathcal{M}}^s}(i)^{-1} \tag{12}$$

5 Evaluation

5.1 Dataset

Our evaluation dataset consists in 8 Full HD, 30 fps RGB video sequences, acquired outdoor at daylight, which include multiple mobile vehicles except 'C0'. 'C0', 'C1' 'C2' and 'C4' were acquired with a light drone on overhead viewpoint, equipped with a GoPro camera. The 4 minutes long sequence 'C1' and 'C4' feature few vehicles and only light motion in overhead configuration; 'C0' is a 15 seconds long subset of 'C1' where all objects are static. To challenge the robustness of our approach, the dataset includes the 4 minutes long sequence 'C2', where the human operator performs two fast clockwise 180 degrees rotation around the vertical axis. 'M1', 'M2', 'M3' and 'M4' are successive 15 minutes long videos which were acquired with a tethered drone equipped with a CMOS camera, at approximately 50 meters above ground in stationary flight. The sequence 'M4' is subject to heavy motion, probably caused by windy conditions. None of those videos displayed significant rolling shutter issues. Figures 1.a and 1.b provide examples from sequences 'M4' and 'C2'.

5.2 Implementation

This approach was implemented using computationally lightweight algorithms provided by the OpenCV library [4] and C++ language. Both $\tilde{\mathcal{M}}^s$ and $\tilde{\mathcal{M}}^r$ were estimated on a sparse image representation basis using the KLT approach.

The image is first resized to 576x324 (30% of a 1080p resolution) and set to one-channel grayscale. The first frame was adopted as the reference frame for the tested videos. $\tilde{\mathcal{M}}^r(0,i)$ (resp. $\tilde{\mathcal{M}}^s(i-1,i)$) was estimated by extracting 200 Shi-Tomasi corners [14] on Im_0 (resp. Im_{i-1}), further tracked on Im_i using the Lukas and Kanade Pyramidal Optical Flow (LKPOF) algorithm [3]. A Least Square Regression (LSR) was used for the motion estimation matrix solving. We used a Kalman Filter [17] on the four motion estimation parameters (t_x, t_y, α, s) independently for the implementation of $\mathcal{F}$ in Eq. (12). The LKPOF algorithm being sensitive to its initialization, we used the KF prediction as the initialization for all of the tracked points locations on the registration part.

The experimentation was carried out on a 2.5 Ghz Intel Core i7 MacBook Pro with 16 Go DDR3 of memory under High Sierra OS. Under these settings, each frame is processed in less than 16 millisecond using CPU operation only, enabling real time applications and leaving space for further processing analysis.

5.3 Evaluation protocol

To show the benefits of the proposed combination, we have compared it with different combinations of its elementary components. The following settings were tested:

- Raw: the original, unprocessed video.
- StabilizationKalman: the video stabilized by the algorithm described on Eq. (5), using the same computation of $\tilde{\mathcal{M}}^s$ and filter as described in section 5.2.
- RegistrationLastPos: the video registered by the algorithm described as $\tilde{\mathcal{M}}^r$ in section 5.2, with the registration proposed at frame $i-1$ as an initialization for the registration of frame i.
- RegistrationKalman: the video registered by the algorithm described as $\tilde{\mathcal{M}}^r$ in section 5.2, with a KF set as described in section 5.2 for both the initialization and the filtering.
- Ours: the proposed method.

Evaluating a stabilization and registration algorithm in our context is a delicate task, since we do not benefit from ground-truth data about the actual camera movements or the image content on our dataset.

To quantify the registration performances of our approach, we propose to track a set of feature points from the reference frame to the current frame, using the same tracker setting as in section 5.2 for $\tilde{\mathcal{M}}^r(0,i)$. The median displacement of all tracked reference points was used as a measure of registration quality. This measure, denoted as frame displacement (fd), was computed on each frame independently.

To quantify the stabilization performances of our approach, we propose to calculate the mean absolute difference of pixel grayscale values between two consecutive frames, over the length of a video sequence. To avoid parts of the image where we had no data, this measurement was performed on the overlapping regions between both consecutive images ($mpvd$).

5.4 Results

The first property that we wanted to quantify is whether our solution is capable of keeping registered to a constant viewpoint. We computed the proposed mean *fd* over the whole course of tested sequences (Table 1). Results emphasize the idea that a stabilization technique, such as StabilizationKalman, is not designed to guarantee a constant viewpoint over the course of a video. On videos 'C0', 'C1' and 'M2', which are subject to little camera motion, all tested registration methods (RegistrationLastPos, RegistrationKalman) and the proposed method perform very similarly. RegistrationKalman suffers from inertia, which degrades its performance, eventually leading to being badly registered during several hundreds of frames on 'C2'. On all tested videos, the difference between the proposed method and the better evaluated registration method is well within subpixel range. This suggests that the proposed algorithm effectively preserves the registration performances of its base component.

Settings	M1	M2	M3	M4	C0	C1	C2	C4
Raw	17.43	17.91	14.81	27.1	11.19	11.79	20.04	9.15
StabilizationKalman	18.83	18.11	15.18	25.96	13.78	13.40	20.28	10.46
RegistrationLastPos	**4.85**	<u>3.84</u>	**6.51**	**11.85**	**0.74**	**1.29**	**4.70**	**1.09**
RegistrationKalman	6.21	**3.83**	7.33	12.49	0.96	1.40	14.07	1.19
Ours	<u>4.86</u>	3.85	<u>6.55</u>	<u>11.87</u>	<u>0.75</u>	<u>1.30</u>	**4.70**	<u>1.12</u>

Table 1. Mean frame displacement (*fd*) on our test sequences (pixels). In bold: the best performance ; underlined: the second best performance.

The second evaluation focused on assessing the stability properties of the different methods based on the *mpvd* values (Table 2). The assumption here is that on stable sequences, only mobile objects should cause pixel values to change significantly from one frame to the next one. On the other hand, jitter would cause pixel values to change suddenly over significant parts of the image, including the background. Stability here is assessed by the lowest possible *mpvd* value. This is verified in our experiment. In general, the poorest performance is visible on the original image, which is unstable. RegistrationLastPos, where jitter occurs despite the image being overall well registered, displays important values. Filtering the output of the registration (RegistrationKalman) significantly improves the results, which shows that this solution was able to tackle most of the jitter issues. On all of the sequences, the better performances are observed with StabilizationKalman, and the proposed combination. The proposed method displays the highest performances because of its ability to keep consistently registered to the same viewpoint.

Settings	M1	M2	M3	M4	C0	C1	C2	C4
Raw	6.80	7.20	7.15	7.25	2.02	2.20	2.71	2.26
StabilizationKalman	**3.24**	<u>3.32</u>	**3.38**	**3.30**	<u>1.49</u>	**1.60**	<u>2.09</u>	<u>1.65</u>
RegistrationLastPos	6.46	4.96	6.54	7.88	2.22	2.39	3.92	2.11
RegistrationKalman	5.37	5.60	5.61	5.80	1.50	1.65	2.21	1.67
Ours	**3.24**	**3.25**	**3.38**	<u>3.41</u>	**1.43**	**1.60**	**1.59**	**1.62**

Table 2. Mean pixel value difference (*mpvd*) between consecutive frames (grayscale value). In bold: the best performance ; underlined: the second best performance.

This quantitative outcome confirms the robustness of the proposed method and the qualitative impression given by visual inspection of the videos [3] . Our proposed approach can be effectively labeled as a jitter-free registration method.

6 Conclusion and perspectives

In this paper, we have addressed the problem of generating a constant viewpoint from videos acquired by stationary UAVs. The camera being subjected to small movements, the view is unstable and poses a problem for applying automatic processing techniques, or long term analysis such as trajectory registration. In this context, we have proposed a generic model to describe the inherent error of motion estimation algorithms. We have used it as the foundation on how to combine registration and stabilization techniques into one single hybrid method. The method is real time and online. It prevents both jittery and drifting behavior, even in the presence of multiple mobile objects. Results show that it retains the better properties out of the tested stabilization and registration techniques.

Further work will focus on two main aspects. One of them is to investigate how to handle situations where linear 2D-rigid warping is inappropriate, for instance when significant parallax is observed. The second aspect is how to update the reference image during the course of a day. This should enable us to better cope with appearance changes on the background, such as lighting conditions, which is a common problem during video surveillance applications.

Acknowledgements. This work was funded by AURA region (Pack Ambition Recherche 2017). Station'air project, number 1701104601-40893.

References

1. Abdelli, A.: Recursive motion smoothing for online video stabilization in wide-area surveillance. In: Big Data and Smart Computing (BigComp), 2016 International Conference on. pp. 40–45. IEEE (2016)

[3] `http://liris.univ-lyon2.fr/~pi/stationair/`

2. Aguilar, W.G., Angulo, C.: Real-time model-based video stabilization for microaerial vehicles. Neural processing letters **43**(2), 459–477 (2016)
3. Bouguet, J.Y.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corporation **5**(1-10), 4 (2001)
4. Bradski, G., Kaehler, A.: Opencv. Dr. Dobbs journal of software tools **3** (2000)
5. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(10), 1858–1865 (2008)
6. Grundmann, M., Kwatra, V., Essa, I.: Auto-directed video stabilization with robust l1 optimal camera paths. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 225–232. IEEE (2011)
7. Guo, H., Liu, S., He, T., Zhu, S., Zeng, B., Gabbouj, M.: Joint video stitching and stabilization from moving cameras. IEEE Transactions on Image Processing **25**(11), 5491–5503 (2016)
8. Liu, F., Gleicher, M., Wang, J., Jin, H., Agarwala, A.: Subspace video stabilization. ACM Transactions on Graphics (TOG) **30**(1), 4 (2011)
9. Liu, S., Yuan, L., Tan, P., Sun, J.: Steadyflow: Spatially smooth optical flow for video stabilization. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. pp. 4209–4216. IEEE (2014)
10. Lupashin, S., D'Andrea, R.: Stabilization of a flying vehicle on a taut tether using inertial sensing. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2432–2438 (Nov 2013). https://doi.org/10.1109/IROS.2013.6696698
11. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017)
12. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4104–4113 (2016)
13. Shen, H., Pan, Q., Cheng, Y., Yu, Y.: Fast video stabilization algorithm for uav. In: Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on. vol. 4, pp. 542–546. IEEE (2009)
14. Shi, J., et al.: Good features to track. In: Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on. pp. 593–600. IEEE (1994)
15. Tauro, F., Porfiri, M., Grimaldi, S.: Surface flow measurements from drones. Journal of Hydrology **540**, 240–245 (2016)
16. Wang, M., Yang, G.Y., Lin, J.K., Zhang, S.H., Shamir, A., Lu, S.P., Hu, S.M.: Deep online video stabilization with multi-grid warping transformation learning. IEEE Transactions on Image Processing **28**(5), 2283–2292 (2019)
17. Welch, G., Bishop, G.: An introduction to the kalman filter. Tech. rep., Chapel Hill, NC, USA (1995)
18. Yang, Z., Dan, T., Yang, Y.: Multi-temporal remote sensing image registration using deep convolutional features. IEEE Access **6**, 38544–38555 (2018)