Feature Trajectories Selection for Video Stabilization

Wilko Guilluy
L2TI, Université Paris 13
Villetaneuse, France
wilko.guilluy@univ-paris13.fr

Laurent Oudre
L2TI, Université Paris 13
Villetaneuse, France
laurent.oudre@univ-paris13.fr

Azeddine Beghdadi
L2TI, Université Paris 13
Villetaneuse, France
azeddine.beghdadi@univ-paris13.fr

***Abstract*—In this paper we present a new method to select the most relevant feature trajectories that could be used in video stabilization algorithms.The main objective is to identify the most appropriate trajectories of the video that could be used for the estimation of the camera motion. We use duration and motion criteria with a global rather than local approach for outlier rejection, thus avoiding the need for a known motion model. The performance of the proposed method is evaluated on several real videos and compared to the state-of-the-art using some intuitive subjective and objective criteria.**

***Index Terms*—video stabilization, video processing**

I. Introduction

Recent years have seen a marked increase in the production of amateur videos, notably due to the development of camera-equipped phones. While the quality of hand-held cameras has greatly improved, there are still noticeable differences between the perceptual quality of amateur and professional video captures. This is mainly due to camera motion during the video capture. Though some software and hardware solutions such as optical stabilizers [1], embedded in video cameras in order to deal with annoying jitters and other annoying camera motion effects have been proposed, video stabilization is still a very challenging problem [2]. These involuntary movements are source of visual discomfort for viewers and may affect the performance of some video processing and analysis tasks [3]. Digital video stabilization is therefore needed to smooth out these artifacts in order to improve the video quality.

Video stabilization operates in several interdependent steps. First, the video motion field is estimated using a frame-to-frame matching process. This estimation can be performed by tracking a set of salient features or points of interest in the successive frames. Several feature points trackers have been proposed in the literature. The most popular are SIFT, SURF or KLT [4]. The position of a given feature point throughout the video forms a feature trajectory, that represents the movement of an object in the video. Secondly, the original moving camera path is computed by using the estimated two-dimensional flow field. Usually a 2D or 3D motion model is used and the camera parameters are computed by solving linear equations. The camera path is then corrected and smoothed to obtain more coherent and smooth movement. Finally, a video restoration process based on the estimated camera path is used.

The estimation of the 2D or 3D camera parameters from feature trajectories is a tricky process since not all movements present in the video give information on the camera motion. While static objects are only affected by camera-induced movements, other objects undergo displacements that are caused by both the camera motion and the movements of the object in the scene. These moving objects need to be separated from the others and removed in order to compute the correct camera path. The most common approach to handle this issue is to use the RANSAC algorithm [5] which is based on a parametric motion model between two successive frames (e.g. affine transform, homography...) [6]–[8]. The feature points that do not fit the model are considered as outliers and removed. In RANSAC, outliers are detected by thresholding the projection error. The threshold can be adapted to the considered frames [9], or fixed using other approaches such as the number of false alarms or negative log-likelihood [10]. Other considerations such as temporal or spatial constraint have been used to improve the effectiveness of RANSAC [6], [14]. By assuming low camera movements, simple strategies can be implemented by removing feature points with a velocity above a given threshold [11]. Alternatively, neighbourhood information could be exploited to remove undesirable moving objects under the assumption of locally smooth motion vector field. The dense optical flow assumption could be then used to remove spurious movements by thresholding the motion gradient [12]. Delaunay triangulation can be used to establish neighbourhood constraints, removing points whose motion differs from those of points lying along an edge of the triangulation [13].

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

In this article, we propose a novel approach to assess and select the best feature trajectories to use in the camera motion estimation for video stabilization. Unlike standard approaches used for the selection of feature trajectories, we analyse the movement of the feature trajectories through all frames and compute a global weight by considering multiple criteria such as movement and duration. Section II presents the proposed method for feature trajectories selection. Section III is devoted to the performance evaluation, along with results on several videos and a comparison with the state-of-the-art. The last section provides some concluding remarks and some open problems and perspectives.

II. Feature Trajectories Selection

First, let us consider a video corrupted by camera movements, from which feature trajectories are extracted using the KLT tracker [4]. This tracker detects interest points and tracks them throughout the video to form feature trajectories. Let $\mathbf{z}_i[t] = (x_i[t],\ y_i[t])^\dagger$ denote the position of the i^{th} feature point at frame t. The instantaneous velocity of this feature point is denoted $\dot{\mathbf{z}}_i[t] = \mathbf{z}_i[t+1] - \mathbf{z}_i[t]$. Since trajectory i might not last for the whole video duration, let us define t_i^{start} and t_i^{end}, as the starting time and end time, respectively, of the i^{th} trajectory.

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

1) First, we analyse each feature trajectory on a local time-window, in order to account for its duration and movement properties. More specifically, we define two local weights $w_i^d[t]$ and $w_i^m[t]$ within the range $[0,1]$ that rank trajectory i according to its duration and its adequacy with the movements observed on a time-window centred on frame t.
2) Then, we combine all local weights $w_i^d[t], w_i^m[t]$ in order to form a global trajectory weight w_i that accounts for the phenomenon observed during the whole duration of the trajectory.
3) We select the feature trajectories with the largest weights w_i to estimate the camera motion parameters.

A. Duration characteristics of features trajectories

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

Here, rather than using a hard threshold as done by Liu & al [14], we use a time window of length $2N_w + 1 = 31$ centred on frame t, and to compute a duration weight $w_i^d[t]$ that accounts for the local duration of trajectory i. This weight is defined as:

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

This weight is within the range $[0,1]$ and corresponds to the percentage of frames within the temporal window of interest for which trajectory i is defined. It is a local score that provides a temporal assessment of the reliability of the feature trajectory.

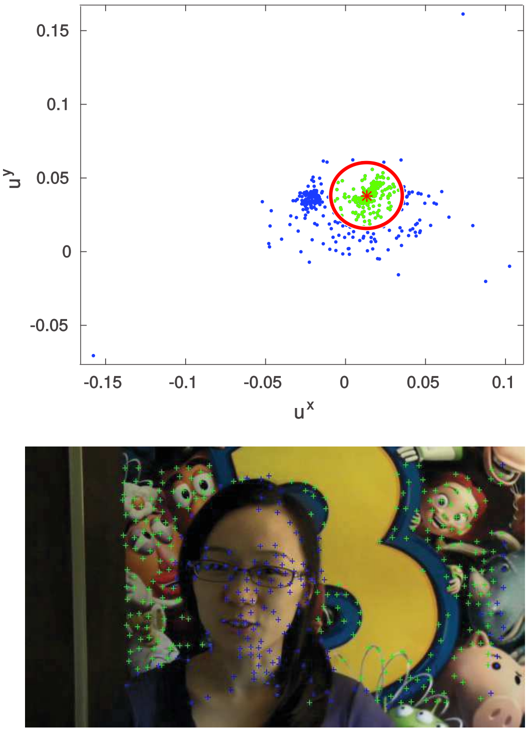

Fig. 1. Example on the *close_person* video (frame 100). All trajectories belonging to the time-window of interest are projected in the $(\mathbf{u}^x, \mathbf{u}^y)$ plane. The majority of the contributions aggregate around the red point (mode of the 2D histogram). When selecting only feature points close to this mode (green points), we retrieve feature points from the background that are only corrupted by camera motion. On the contrary feature points far from the mode (blue points) correspond either to the moving woman or to spurious feature points.

B. Motion characteristics of features trajectories

A second criterion related to the motion characteristics of the features trajectories is introduced and used in order to discriminate between static and moving objects in the distorted video. However, without knowledge of the scene characteristics, deriving the most appropriate motion model corresponding to the video is an ill-posed problem. Therefore, instead of using RANSAC [5] or its variants [10], which are based on parametric and geometric models, we propose to identify the dominant movement in the video without assuming any motion model, by using a projection in a low-rank subspace.

We consider a time window of length $2N_w + 1 = 31$ centred on frame t, and form the local velocity matrix $\dot{\mathbf{Z}}[t]$ which contains all instantaneous velocities $\{\dot{\mathbf{z}}_i[\tau]\}_{\tau=t-N_w}^{t+N_w}$ belonging to trajectories that overlap with the time-window of interest. The temporal window is small since the approximation of motion in a low-rank subspace does not hold over long periods. This matrix is then analyzed with a Singular Value Decomposition (SVD) algorithm that handles missing values through an iterative process [15] :

$$\dot{\mathbf{Z}}[t] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\dagger} \tag{2}$$

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

This decomposition has been used to model the camera motion using the right-singular vectors [14]. We instead use it to analyse the components which influence the feature trajectories, using the left-singular vectors.The largest singular value λ_1 in $\mathbf{\Sigma}$ captures the information corresponding to the dominant movement that is localized within the time-window of interest. The first left-singular vector $\mathbf{u}_1$ in $\mathbf{U}$ corresponds to the contribution of each trajectory to this dominant movement. Intuitively, all feature trajectories belonging to static objects or background, should have similar contributions to this dominant movement. On the contrary, moving objects or pixels should have different contributions and thus be identifiable. First, $\mathbf{u}_1$ is decomposed by taking every other line and forming $\mathbf{u}^x$ and $\mathbf{u}^y$, with $\mathbf{u}^x$ containing the parameters describing the parameters of the horizontal movements and $\mathbf{u}^y$ the parameters of the vertical movements. Figure 1 illustrates this process: feature points affected only by camera motion tend to aggregate in the $(\mathbf{u}^y, \mathbf{u}^x)$ plane. By detecting the mode (u_*^x, u_*^y) of the 2D-histogram (16×16) of these contributions, we can define a movement weight $w_i^m[t]$ that accounts for the distance (in the $(\mathbf{u}^y, \mathbf{u}^x)$ plane) of trajectory i to the detected mode:

$$w_i^m[t] = e^{-\gamma\left[(u_i^x - u_*^x)^2 + (u_i^y - u_*^y)^2\right]} \tag{3}$$

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

C. Combination process

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

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

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

$$w_i = \frac{1}{t_i^{end} - t_i^{start} + 1} \sum_{t=t_i^{start}}^{t_i^{end}} w_i^d[t] \times w_i^m[t]. \tag{4}$$

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

This weighting method can be used in several strategies for feature trajectories selection. In this work, we remove all trajectories whose weight w_i is lower than a threshold λ. This threshold is set such as there is always a minimum number N_s of trajectories present in each frame. We tested different values using an objective criterion based on resolution loss (detailed in section III) and chose the value $N_s = 40$ which provided the best results on the 15 videos of the dataset. Since few trajectories are needed, rejecting many viable trajectories is preferable to accepting a single outlier, however retaining too few trajectories seems to result in over-fitting the motion model to a small area containing the retained features.

III. Results and discussion

The selection process introduced in this article can be seen as a pre-processing step that can be used in any video stabilization method that relies on feature trajectories. This step can be evaluated independently by visual inspection of the selected trajectories or as part of a video stabilization process. Figure 2 presents the selected trajectories for the video *close_person*. In this video, the woman is moving in front of a static background that is only corrupted by camera movements. The selection process successfully extracts $N_s = 40$ trajectories belonging to the static background and all those belonging to the moving foreground are removed. Additional results on five different videos can be found on our webpage[1]: in all tested videos, the trajectories belonging to moving objects are correctly rejected by our method.

In the following, we investigate the relevance and the impact of the selection step within the video stabilization process. To that end, we propose to plug our pre-processing step into a standard video stabilization method, called Local Linear

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

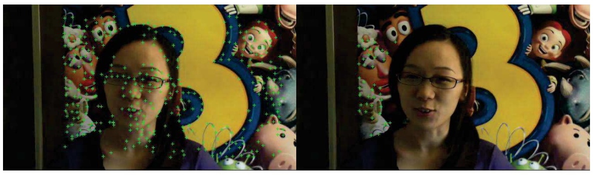

Fig. 2. Example on the *close_person* video (frame 100). On the left frame are all trajectories detected by the KLT tracker, and on the right are the selected trajectories. All trajectories corresponding to the moving woman have been removed.

Matrix-Based smoothing (LLMB) [16]. First, we estimate the 2D affine model $H_{t,t+1}$ between two successive frames t and $t+1$

$$H_{t,t+1} = \begin{pmatrix} 1+a_{11} & a_{12} & T_x \\ a_{21} & 1+a_{22} & T_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (5)$$

These transformations are computed by solving a least-square problem from the feature trajectories retained by our trajectory selection algorithm. Then, these transformations are accumulated and smoothed using a Gaussian filter (with $\sigma = 50$). We denote the accumulated transforms $\bar{H}_t$ and the smoothed transforms $\tilde{H}_t$. The difference between the accumulated and smoothed transformations defines an inverse transform $\hat{H}_t^{-1} = (\tilde{H}_t - \bar{H}_t + I_3)^{-1}$ that can be applied to frame t in order to diminish the camera movements. The resulting stabilized video is finally cropped to avoid undefined regions. The flowchart of our stabilization framework is presented on Figure 3.

The proposed stabilization framework is compared to a RANSAC based video stabilization method and the Youtube stabilizer [17]. While our method selects reliable trajectories to compute the 2D transforms, the RANSAC based approach considers all trajectories and selects the motion parameters that results in the highest number of trajectories being considered inliers. The 2D affinity transforms are then processed with the same LLMB stabilization method [16]. In the case of the Youtube stabilizer, the video stabilization is performed using cinematographic criteria after removing the local outliers [17].

These methods have been tested on fifteen videos presenting different challenges for video stabilization, such as large moving object or depth differences [14]. Some representative exemples of these results is available on our webpage[1]. Figure 4 presents a screen-shot of the results obtained by the three methods for the *14_object* video which depicts a train leaving a station. Interestingly, the motion of the train is interpreted by both the RANSAC algorithm and the Youtube stabilizer as being part of the camera movement. The RANSAC algorithm seeks a compromise between the parameters of the background movement and the train motion, causing distortions in the video, that are outlined in green in the figure 2. The Youtube stabilizer avoids spatial distortions but causes artificial camera motion in an attempt to stabilize the train motion in short bursts before re-centring on the original camera path. These temporal distortions are visible on the video but are unfortunately impossible to display on still image frames. We recommend the readers to refer to our website for better resolution and display. Note that in both cases the attempt to correct the movements of an object causes the RANSAC and Youtube algorithm to stray further than necessary from the original camera path, causing a loss in resolution. This loss is easily visible for RANSAC and our method since cropped areas are visible in black, but it can also be seen in the Youtube result. Note how the door on the left has been cropped compared to the other exemples. Results on other videos show similar effects: our method allows robustness in the presence of large moving objects for which both the RANSAC algorithm and the Youtube video editor fail.

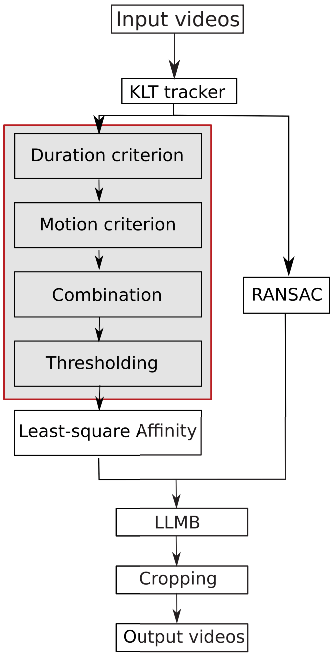

Fig. 3. Flowchart of the stabilization framework. The selection process introduced in this article is displayed in gray.

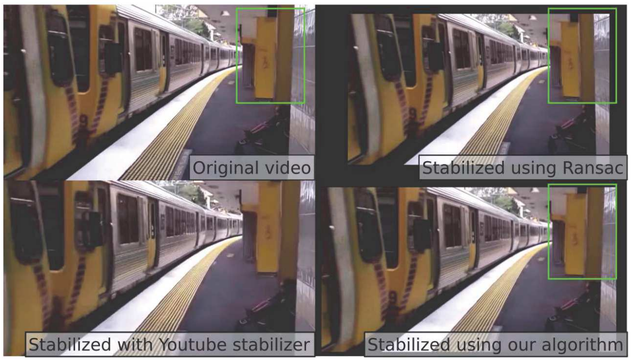
Original video
Stabilized using Ransac
Stabilized with Youtube stabilizer
Stabilized using our algorithm

Fig. 4. Example on the *14_object* video (frame 68). On the upper left is the original video, on the upper right the result of the RANSAC algorithm, on the lower left the results of the Youtube stabilizer and on the lower right the result of our algorithm. Cropped areas are displayed in black for RANSAC and our method. Youtube stabilizer automatically resizes the video.

TABLE I
MEAN PERCENTAGE OF UNDEFINED AREA BEFORE CROPPING

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

Objective evaluation of stabilization methods is a challenging task, especially for real videos, for which the ground truth camera path is not available. Some authors [16] have recently introduced an unsupervised and objective criterion for the evaluation of video stabilization, which is based on resolution loss. The idea is to compare the percentage of empty regions obtained by several methods with the exact same level of stabilization (σ value). Indeed, different degrees of stabilization naturally lead to differences in resolution loss. However by comparing this percentage on videos treated with the same stabilization method, we can judge whether our trajectory selection helps in limiting the resolution loss. Intuitively, this criterion illustrates how close the estimated transformations using our trajectory selection are to the true camera movement compared to the transformations computed using RANSAC. The results in terms of mean percentage of undefined area before cropping, obtained on 15 videos when using the RANSAC based method and the proposed scheme are summarized in Table I. Unfortunately, the results are not available for Youtube stabilizer as it uses different stabilization and cropping strategies. Table I shows that using our trajectory selection reduces the average undefined area (-1.5% in average), which leads to better resolution after cropping. In particular, scenes containing large moving objects greatly benefit from our method, making it possible to obtain a strong stabilization while keeping acceptable video resolution.

IV. Conclusion

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

References

[1] J.-M. Koo, M.-W. Kim, and B.-K. Kang, “Optical image stabilizer for camera lens assembly.” Patent, 2009. US Patent 7,489,340.

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

[3] S. Megrhi, M. Jmal, W. Souidene, and A. Beghdadi, “Spatio-temporal action localization and detection for human action recognition in big dataset,” *Journal of Visual Communication and Image Representation*, vol. 41, pp. 375–390, 2016.

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

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

[6] A. Goldstein and R. Fattal, “Video stabilization using epipolar geometry,” *ACM Transactions on Graphics*, vol. 31, no. 5, pp. 126:1–126:10, 2012.

[7] S. Liu, L. Yuan, P. Tan, and J. Sun, “Bundled camera paths for video stabilization,” *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 78:1–78:10, 2013.

[8] S. Jeon, I. Yoon, J. Jang, S. Yang, J. Kim, and J. Paik, “Robust video stabilization using particle keypoint update and l1-optimized camera path,” in *Sensors*, 2017.

[9] M. R. Souza and H. Pedrini, “Digital video stabilization based on adaptive camera trajectory smoothing,” *EURASIP Journal on Image and Video Processing*, vol. 2018, p. 37, May 2018.

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

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

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

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

[14] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, “Subspace video stabilization,” *ACM Transactions on Graphics*, vol. 30, no. 1, pp. 4:1–4:10, 2011.

[15] N. Srebro and T. Jaakkola, “Weighted low-rank approximations,” in *Proceedings of the International Conference on Machine Learning (ICML)*, (Washington D.C, USA), pp. 720–727, 2003.

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

[17] M. Grundmann, V. Kwatra, and I. Essa, “Auto-directed video stabilization with robust l1 optimal camera paths,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Colorado Springs, CO, USA), pp. 225–232, 2011.