A Joint Spatial-Temporal 3D Video Stabilization Algorithm

Jie Zhou[1], Zhixiang You[1], Ping An[1(✉)], Xinliang Wu[2], and Tengyue Du[1]

[1] School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China
anping@shu.edu.cn
[2] China Aeronautical Radio Electronics Research Institute, Shanghai 200233, China

Abstract. This paper presents a 3D (Three dimensional) video stabilization algorithm combined with a joint spatial and temporal strategy. On the temporal axis, SURF (Speeded-Up Robust Features) are extracted from the consecutive frames and then motion parameters are estimated, with which we calibrate and compensate the video frames after smoothing the motion parameters using Kalman filtering. Then, on the spatial axis, a histogram statistics method based on the extracted features is applied to detect the vertical parallax between the two views. Adjustments are implemented only when the parallax is larger than the safety threshold, which is conducted through subjective assessment, to maintain the consistency of 3D videos. The experimental results have shown that the proposed method is effective to reduce the vertical instability and inconsistency between binocular views and improve the quality and comfortableness of 3D videos.

Keywords: 3d video stabilization · SURF feature · Motion parameters · Kalman filter

1 Introduction

With the development of image sensing technology and mobile computing power in last years, the video acquisition is gradually shifting to hand-held devices, which allow many amateurs to capture personal videos easily despite that a considerable number of these videos are affected by unwanted camera shakes and jitters, leading to low quality video and visual uncomfortableness. The hardware methods make use of mechanical principle such as fixing the cameras on the Steadicam, which are too expensive or inconvenient for amateurs.

On the other hand, digital video stabilization is the process of removing undesirable shakes and jitters and compensating the video sequences, which only takes advantage of the information of video frames and does not need camera motion information additionally. So that it is not expensive and has high stability precision. Furthermore, this technology can be implemented in real-time. In recent years, digital video stabilization is widely applied to improve the quality of videos captured by hand-held cameras, video

X. Yang and G. Zhai (Eds.): IFTC 2016, CCIS 685, pp. 72–82, 2017.
DOI: 10.1007/978-981-10-4211-9_8

monitoring based on the motion platform, vehicle-mounted mobile video stabilization and robot navigation.

Video stabilization system is generally divided into two basic parts: motion estimation and motion smoothing. The motion estimation system generally makes an attempt to establish the global motion between nearby frames in a video. To reach this goal, there are two main current strategies: optical flow estimation and feature matching methods. In [1], the motion between successive frames was estimated by optical flow. In [2], a special optical flow called SteadyFlow was proposed. Different from the optical flow, the SteadyFlow has strong spatial smoothness, so that the pixel profiles from the SteadyFlow can be smoothed to stabilize the video. On the contrary, the latter method makes use of the features between successive frames to track the motion. In [3], the video stabilization algorithm tracked the Scale Invariant Feature Transform features to estimate the interframe motion. Then a modified Iterative Least Squares method reduced the estimation errors in order to improve video stability. In [4], a full-frame video stabilization algorithm based on SIFT feature matching was presented. Firstly, SIFT features were extracted to define the affine of 2D perspective warp between nearby frames. Finally, a temporal domain filter was applied to every frame to remove the high frequency components that are thought as noise or undesired camera jitters. The algorithm in [5] also extracted the SIFT features to track the video motion. With the regard to motion smoothing, a considerable number of methods have been proposed. In [6], Matsushita et al. averaged some affine matrices of neighboring frames as new transformation. In [5], Mengsi et al. firstly filtered the global motion by means of Kalman filter and an ideal low-pass filter with the Hanning window. The drawback of [5] is that the filter was not adaptive and the cutoff frequency should be set before filter processing. In [7], it compared Kalman filter and least square fitting in respect of motion smoothing and drew a conclusion that the fitting method performs better than Kalman filter, while Kalman filtering is more suitable for real-time environments because it needs only one observation data from the previous state. Liu et al. presented a joint subspace stabilization method for 3D video in [11], they jointly constructed a common subspace from the left and right video and it was used to stabilize the two videos simultaneously. The method achieved high-quality stabilization for 3D video. But it had difficulties to handle dominating scene motion, excessive shake and strong motion blur. In [12], feature points were tracked to model stabilized camera motion, and then projected the tracked points onto the stabilized frames using epipolar point transfer technology and image-based frame warping. The algorithm was robust to ambiguities and degenerate camera motion, but it could not deal with strong camera jitters and scene where only little features could be detected.

This paper presents a stabilization method for binocular 3D video combined with both spatial domain and temporal domain. In the temporal domain, for each 2D video, the interframe motion is estimated by tracking the SURF features through successive frames and Kalman filtering is applied to remove the high frequency components (impulsive jitters). Finally, we use a low-pass filter to improve the stabilization performance and fill the miss pixel in order to protect the original resolution. On the other hand, we adjust the vertical disparity between two views in the spatial domain. To speed up

the algorithm, we establish a vertical parallax histogram for the extracted features and adjust the vertical parallax according to the histogram.

The rest of this paper is organized as follows: In Sect. 2, the temporal method for video stabilization based on SURF feature matching is presented. In Sect. 3, the spatial method is described in detail. And the whole algorithm combined with Sects. 2 and 3 is proposed in Sect. 4. Experimental results are given in Sect. 5. Finally, the paper draws the conclusions in Sect. 6.

2 Video Stabilization

This section introduces the video stabilization via feature tracking and hybrid filtering. Our approach uses SURF keypoints as features for tracking interframe motion of each 2D view respectively. SURF features are extracted from two consecutive frames and then these features are required to be matched correctly so as to estimate the motion. However the initial matching cannot give correct information how the current frame moves relatively to the previous. Effective verification procedure should be applied to discard wrong matches.

This scheme assumes that the first frame is stable and take the previous frame as the reference frame to stabilize the current frame. The process doesn't terminate until the last frame is stable.

2.1 Motion Estimation

Paper [3] used Euclidean distance between descriptors' vectors and a distance ratio to refine the feature matches. For every keypoint in the previous frame, there are two candidate matching points in the current frame, which are the closest and the second closest points. The distance ratio means the ratio of closest distance to the second closest one. The reliable match is that the closest distance is small when the second closest distance is larger. As a result, the ratio should be small. On the other hand, when these two distances are close to each other and the ratio is close to one, we can't determine which match is more reliable. We check the ratio against the predefined threshold. If the ratio is smaller than the threshold, the match is reliable. On the contrary, if the ratio is larger, the match should be discarded.

Although the distance ratio can improve the reliability of matches, there is a problem that a point A from the previous frame can match a point B well, but any matching point can not be found in the previous frame for the point B. To solve this problem, we use a symmetrical strategy and two sets of features are verified whether they are matched each other. Finally the RANSAC algorithm [17] is applied to further verify the reliability of feature matches. After the above processing, matches are reliable enough. According to these reliable matches, the video motion can be tracked. However, the video motion should also be smoothed because of the noise in the video.

2.2 Motion Smoothing

As described in the previous section, the motion can be described by homography of 2D perspective transformation. The transformation matrix T can be computed by reliable matches. To smooth the noise in the video, motion filtering are necessary.

First, neighboring frames should be taken into consideration to smooth the series of transformation in the video. Let T_i^j denote the transformation from frame i to j. Here, the index of current frame is assumed as t, and the indices of its 2k+1 neighboring frames can be denoted by $\mathrm{N}_t = \{\mathrm{w} | \mathrm{t} - \mathrm{k} \le \mathrm{w} \le \mathrm{t} + \mathrm{k}\}$. The final transformation T_{final} for smoothing can be denoted as following:

$$\mathrm{T}_{final} = \sum\nolimits_{i \in \mathrm{N}_t} T_i^t * G(||t - i||, \sigma) \tag{1}$$

Where $\mathrm{G}(\mathrm{u}, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{u^2}{2\sigma^2}}$ and $\sigma = \sqrt{k}$. Then the transformation T_{final} is performed on the current frame in order to smooth the motion.

After the perspective transformation, the Kalman filtering algorithm is adopted to remove the camera jitter further. The Kalman filter is known as a recursive filter, which uses a series of measurements over time and produces estimates. Figure 1 shows the structure of Kalman filter.

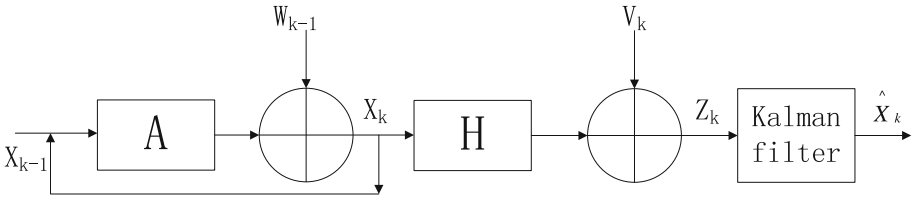


Fig. 1. Kalman filter structure

In Fig. 1, matrix A is the state transformation model between time $k-1$ to k, and W_{k-1} is the system noise obeying Gaussian distribution and its covariance matrix is Q. The current state of the system can be described as follows:

$$X_K = AX_{k-1} + W_{k-1} \tag{2}$$

Z_k is the measurement of state X_k at time k, and matrix H is the measurement matrix. V_k is the measurement of noise and it also obeys Gaussian distribution, whose mean is zero and covariance matrix is R. The measurement state Z_k is:

$$Z_K = HX_k + V_k \tag{3}$$

The relevant matrix of Kalman filter is set up as follows:

$$\mathrm{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathrm{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The covariance matrix of the system noise Q and the covariance matrix of the measurement noise R are predefined, and we choose the diagonal matrix of 0.01 in this paper. We assume that $\hat{X}_k$ is the estimated value of X_k, and P_k is the posteriori error estimate covariance matrix of X_k. The initial value of $\hat{X}_k$ and P_k can be set as any value.

The operation of Kalman is mainly divided into two parts: prediction and correction.

(1) prediction:

$$\begin{aligned} \hat{X}_{k|k-1} &= A\hat{X}_{k-1|k-1} \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q_k \end{aligned} \tag{5}$$

(2) correction:

$$\begin{aligned} K_k &= P_{k|k-1}H^T\left(HP_{k|k-1}H^T + R_k\right)^{-1} \\ \hat{X}_{k|k} &= \hat{X}_{k|k-1} + K_k\left(Z_k - H\hat{X}_{k|k-1}\right) \\ P_{k|k} &= (I - K_kH)P_{k|k-1} \end{aligned} \tag{6}$$

where K_k is the gain matrix of Kalman. We can get output $\hat{X}_{k|k}$ of each frame through two steps above. Because the jitter is the high frequency component in frequency domain, a low-pass filter is applied to remove the high frequency component. The cut-off frequency of low-pass filter is set to 0.5 Hz.

2.3 Video Completion

Since the process of motion smoothing and low-pass filtering results for the missing pixels, a video completion is required to fill the missing pixels so that the original resolution can be protected. This paper uses an average method and a missing pixel is described as follows:

$$\mathrm{I}_t(\mathrm{m}, \mathrm{n}) = \frac{1}{\mathrm{N}_t} \sum\nolimits_{i \in \mathrm{N}_t} \mathrm{I}_i'(\mathrm{m}, \mathrm{n}) \tag{7}$$

Where I_t is the current frame which is needed to be stabilized and I_i' is the frame that has been stabilized.

3 Parallax Adjustment

In [8], Kooi et al. found that vertical parallax, crosstalk and blur could determine the comfort of viewing 3D video through subjective experiments. The definition of vertical parallax is the vertical distances between the same points from different viewpoints. In this paper, pixels in the left view are taken as the baseline and thus the vertical parallax is computed from pixels in the right view to their counterparts in the left view. In addition to horizontal parallax, vertical disparity should be controlled in a certain range. Large vertical parallax in parallax images should be avoided because it is disadvantageous to stereo vision due to the presence of visual fatigue. The larger vertical parallax is, the deeper the level of visual fatigue will be. From the above, we must remove the jitter or parallax in the vertical direction for better 3D vision effect.

First, we extract the SURF features instead of every pixels in the left and right views. This strategy can reduce the complexity greatly. Compared with commonly used SIFT features, SURF features can be extracted faster. To be precise, extracting SIFT features is three times as fast as SURF, but SURF performs worse than SIFT in terms of scale and rotation transformation. Second, we use FLANN (Fast Library for Approximate Nearest Neighbors) for matching extracted features in two views. To ensure the feature point matching is robust and reliable, RANSAC algorithm is implemented to reject outliers and wrong matches. Not many correct feature matches are needed to represent whole pixels in order to compute the vertical parallax. Third, we compute the vertical parallax of matched feature points respectively and average them to obtain the statistical vertical parallax. Finally, we rectify the vertical parallax according to the third step. The experiments in Sect. 4 show that the vertical parallax of the two viewpoints is almost zero in most frames.

4 Overall Framework

Figure 2 shows the framework of this paper. Disparity rectification and video stabilization are put into a framework. The statistics vertical parallax is computed for each frame and it will be compared with a threshold. If it is less than the predefined threshold, we think the parallax is caused by difference of CMOS or CCD in cameras and we will apply parallax rectification to rectify the parallax. On the other hand, if the vertical parallax is larger than the threshold, which maybe created by camera jitter, so that the video stabilization is performed.

5 Experimental Results

The experimental system is implemented in Visual C++ 2013, and the resolution of the experimental video is 1920×1080. Because of the deformation of the holder, the inconsistency of CCD or CMOS or inaccuracy crafts, these factors will bring vertical disparity of different level. Vertical disparity will influence the quality of 3D images and large disparity will produce ghosting. In addition, vertical parallax can lead to keystone distortion or other geometric distortion, which will affect the comfort of 3D images. In

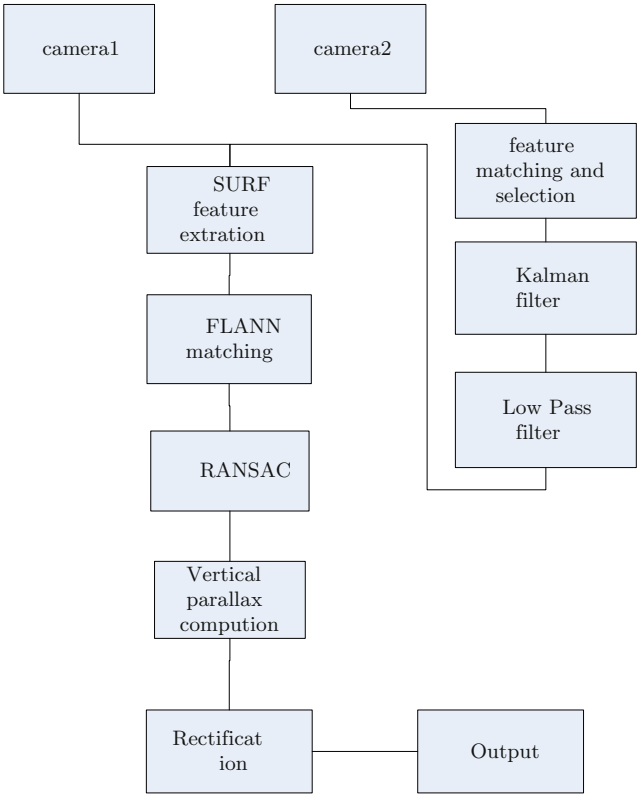


Fig. 2. Framework of the proposed algorithm

view of the disparity, we have done sixteen groups of subjective experiments. Figure 3 is two groups. The horizontal ordinate represents the pixels difference between two views and the vertical ordinate is the subjective scores from fifteen experimenters. The higher scores are, the better comfort viewers can get. In these subjective experiments, we conclude that vertical disparity of more than forty pixels will bring uncomfortableness and the scores drop sharply. The aim of our method is to reduce vertical disparity which caused by jitter motion or any other cue.

Figure 4(a) shows the SURF feature matching and Fig. 4(b) presents the matching result refined by RANSAC. Comparing the two figures, some wrong matches in Fig. 4(a) have been discarded and remain the reliable ones. Correct feature matches can track camera motion more robust.

After the reliable features matches are captured, a hybrid filtering method (Kalman and low-pass filter) is applied to the jitter video. Figure 5 shows the vertical parallax after the video stabilization algorithm is performed, where x-axis is the number of frames and y-axis means the video motion in the vertical direction. The red curve shows the original parallax between the successive frames, while the blue curve is the result after stabilizing the video. The parallax of most frames in blue curve are close to zero.

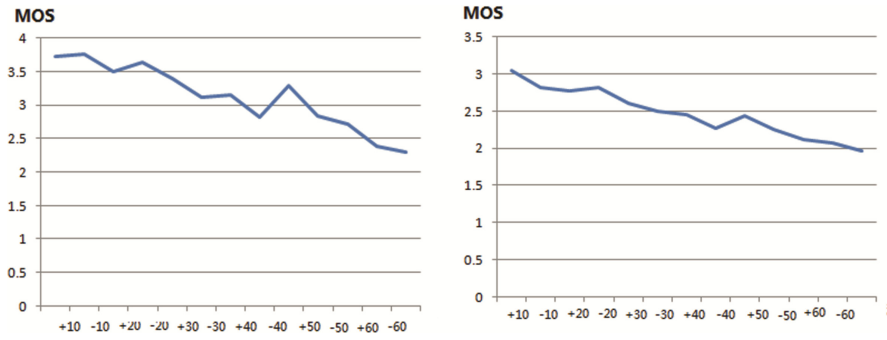
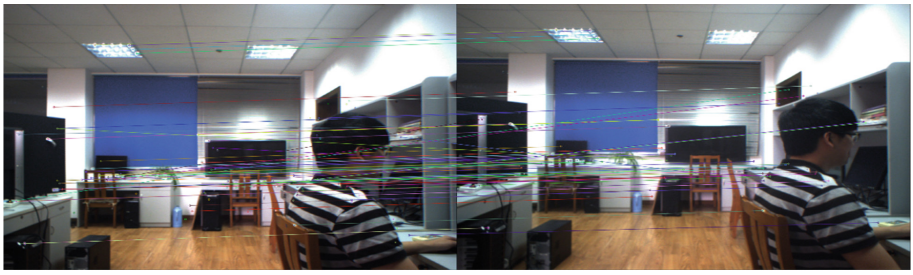
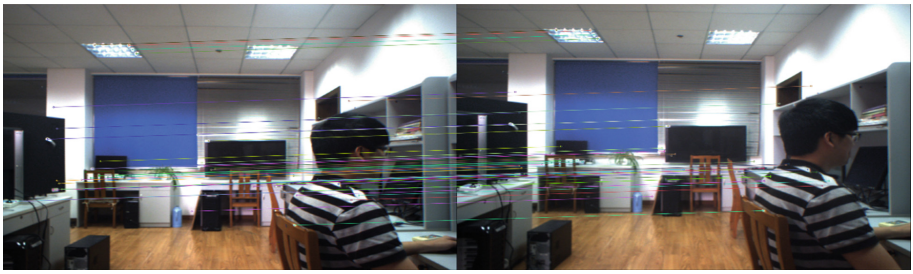


Fig. 3. Subjective experiments of vertical disparity

Figure 6 shows the vertical parallax between original left and right video after corrected feature matching and parallax adjustment, where x-axis represents the number of frames and y-axis means the vertical disparity. The red curve is original parallax between two views and the blue curve is the result of parallax rectification. In this figure, we can see that the rectification process can't deal with large and impulsive vertical parallax.



(a) Initial matching



(b) Refined matching

Fig. 4. Initial and refined SURF feature matching

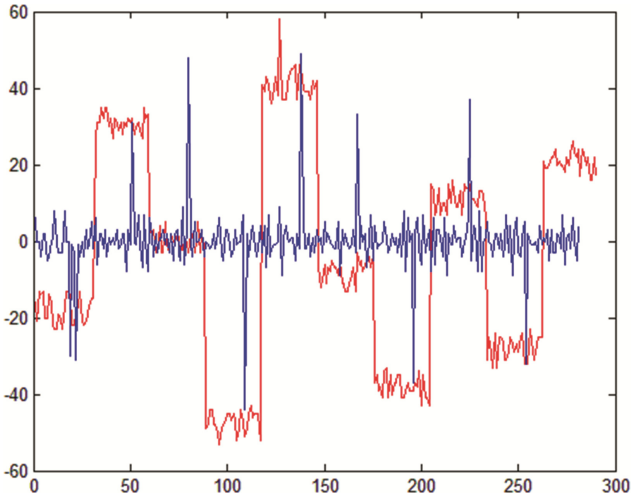


Fig. 5. Vertical parallax curves of video stabilization; the red curve indicates the original parallax and the blue curve is the result after stabilization. (Color figure online)

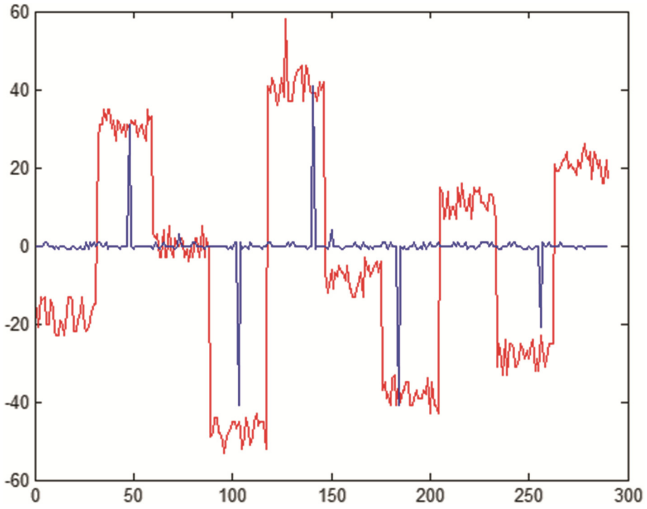


Fig. 6. Vertical parallax curves of parallax rectification; the red curve indicates the original parallax and the blue curve is the result after the parallax adjustment. (Color figure online)

To remove such large and impulsive vertical parallax, the strategy is that the video should be stabilized first and parallax adjustment can be operated on the stabilized video. Because the jitters in the vertical direction can also bring vertical parallax. The vertical parallax from jitter motion should be processed first and then the parallax adjustment algorithm is applied to reduce the vertical parallax further.

In Fig. 7, the blue curve is the original vertical parallax and the red curve is the final result of this paper. It shows that our approach can remove the vertical parallax in most frames and smooth the camera motion at the same time.

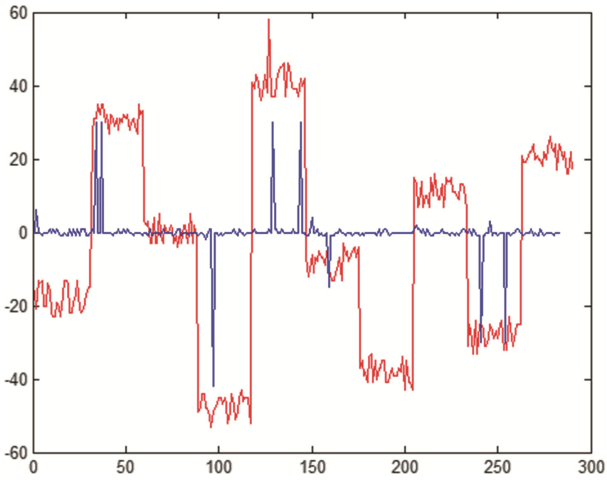


Fig. 7. Vertical parallax curve of final result; the red curve indicates the original parallax and the blue curve is the final result of our algorithm. (Color figure online)

6 Conclusion

In this paper, we present a video stabilization algorithm combined with parallax rectification and feature-based stabilization technology. For smaller parallax, we rectify it directly based on parallax of the feature points. For larger parallax, we take advantage of temporal stabilization strategy for 2D video to smooth the inter-frame motion. Our method has difficulty to handle the strong camera motion and large vertical disparity in few frames, while the algorithm gets good performance in most video frames.

However our temporal method is not a 3D video stabilization, and the next work is to stabilize the video based on the video of the other view. In future work, our algorithm will be speeded up and the stabilized 3D video can be displayed in real time.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China, under Grants 61571285, and U1301257. The work is also supported by the 2016 peak discipline of filmology of Shanghai University.

References

1. Gleicher, M.L., Liu, F.: Re-cinematography: improving the camera dynamics of casual video. In: International Conference on Multimedia, pp. 27–36. ACM (2007)
2. Liu, S., Yuan, L., Tan, P., et al.: SteadyFlow: spatially smooth optical flow for video stabilization. In: Computer Vision and Pattern Recognition, pp. 4209–4216. IEEE (2014)

3. Battiato, S., Gallo, G., Puglisi, G., et al.: SIFT features tracking for video stabilization. In: International Conference on Image Analysis and Processing, pp. 825–830 (2007)
4. Chen, Y.H., Lin, H.Y.S., Su, C.W.: Full-frame video stabilization via sift feature matching. In: Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE (2014)
5. He, M., Huang, C., Xiao, C., et al.: Digital video stabilization based on hybrid filtering. In: International Congress on Image and Signal Processing, pp. 94–98. IEEE (2014)
6. Matsushita, Y., Ofek, E., Tang, X., et al.: Full-frame video stabilization. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005, CVPR 2005, vol. 1, pp. 50–57 (2005)
7. Yu, H., Zhang, W.: Moving camera video stabilization based on Kalman filter and least squares fitting. In: Intelligent Control and Automation. IEEE (2015)
8. Kooi, F.L., Toet, A.: Visual comfort of binocular and 3D displays. Displays **25**(2–3), 99–108 (2004)
9. Liu, W.X., Chin, T.J.: Smooth globally warp locally: video stabilization using homography fields. In: International Conference on Digital Image Computing: Techniques and Applications. IEEE (2015)
10. Salunkhe, A.U., Jagtap, S.K.: A survey on an adaptive video stabilization with tone adjustment. In: International Conference on Computing Communication Control and Automation. IEEE (2015)
11. Liu, F., Niu, Y., Jin, H.: Joint subspace stabilization for stereoscopic video. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 73–80. IEEE (2013)
12. Goldstein, A., Fattal, R.: Video stabilization using epipolar geometry. ACM Trans. Graph. **32**(5), 573–587 (2012)
13. Liu, S., Yuan, L., Tan, P., et al.: Bundled camera paths for video stabilization. ACM Trans. Graph. **32**(4), 96 (2013)
14. Song, J., Ma, X.: A novel real-time digital video stabilization algorithm based on the improved diamond search and modified Kalman filter 91–95 (2015)
15. Pinto, B., Anurenjan, P.R.: Video stabilization using speeded up robust features. In: 2011 International Conference on Communications and Signal Processing (ICCSP), pp. 527–531. IEEE (2011)
16. Mayen, K., Espinoza, C., Romero, H., et al.: Real-time video stabilization algorithm based on efficient block matching for UAVs. In: The Workshop on Research, Education and Development of Unmanned Aerial Systems. IEEE (2015)
17. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)