

# Video Stabilization Using Sliding Frame Window

Keerthan S. Shagrithaya<sup>1</sup>, Eeshwar Gurushankar<sup>1</sup>, Deepak Srikanth<sup>1</sup>,  
Pravin Bhaskar Ramteke<sup>2</sup>, and Shashidhar G. Koolagudi<sup>1</sup>

National Institute of Technology Karnataka, Surathkal, Karnataka, India  
keerthanss9@gmail.com, eeshwarg13@gmail.com, deepakks96@gmail.com,  
ramteke0001@gmail.com, koolagudi@nitk.ac.in

**Abstract.** Shaky videos are visually unappealing to viewers. Digital video stabilization is a technique to compensate for unwanted camera motion and produce a video that looks relatively stable. In this paper, an approach for video stabilization is proposed which works by estimating a trajectory built by calculating motion between continuous frames using the Shi-Tomasi Corner Detection and Optical Flow algorithms for the entire length of the video. The trajectory is then smoothed using a moving average to give a stabilized output. A *smoothing radius* is defined, which determines the smoothness of the resulting video. Automatically deciding this parameter's value is also discussed. The results of stabilization of the proposed approach are observed to be comparable with the state of the art YouTube stabilization.

**Keywords:** Video stabilisation · Feature trajectories · Smoothing radius

## 1 Introduction

With advances in smart-phone technology and the ubiquity of these hand-held devices, every important moment of our lives is captured on video. But these videos are of often poor quality: lacking stabilizing equipment such as tripods, steady-cams, gimbals, etc. They are often shaky and unstable, especially when the videographer, the subject, or both are in motion. To deal with this, affordable techniques that do not require hardware to stabilize such videos are of the essence. Post-processing techniques that need no additional hardware are effective in removing the effects of jerky camera motion, and are independent of the device capturing the video and the subject of the video. The proposed approach is a five step sequential process that covers the different stages of video stabilization from feature extraction and tracking to camera motion estimation and compensation. This method is run on a set of videos and is compared to the state of the art YouTube video stabilizers.

Section 2 discusses the relevant work done in the field of video stabilization. The proposed video stabilization technique is discussed in detail in Sect. 3. Section 4 describes the experiments performed and results obtained. The paper concludes with Sect. 5 which gives the inferences made along with future directions.

## 2 Literature Survey

2D video stabilization methods perform three tasks - motion estimation, motion compensation, and image correction. Feature matching is done using Scale Invariant Feature Transform (SIFT) [1] or Oriented FAST and Rotated BRIEF (ORB) operators [12]. Robust feature trajectories [5] involve using features matched along with neighboring features information. It use SIFT or ORB descriptors to match features between frames to calculate feature trajectories, and then pruning and separating them into local and global motion. The motion inpainting algorithm, which fills frame's unpainted regions by observing motion history [5], serves well for image correction. Another approach modified the optical flow algorithm [3, 8], where pixels were tracked to find which feature trajectories cross it [7]. The approach worked well with videos possessing large depth change, but failed in cases with dominant large foreground objects. Videos having large parallax or rolling shutter effects are challenging to stabilize. To tackle this, a bundle of trajectories were calculated for different subareas of the image and these were all processed together [6]. Applying L1-norm optimization [2] can generate camera paths that consist of only constant, linear and parabolic motions, which do follow cinematography rules. YouTube has since adopted this algorithm into their system.

Most of the approaches discussed here witness a significant difference in performance on videos with an object of interest and ones without, with the latter facing a dip. In this paper, an approach based on [10] is investigated whose performance does not change for videos with objects of interest or ones without.

## 3 Methodology

The proposed approach uses two structures - *transform parameter* and *trajectory*. The first holds the frame-to-frame transformation-  $dx$ ,  $dy$ ,  $da$ . Each represents changes in the x-coordinate, y-coordinate and rotational change respectively. The latter structure stores the positions and angle of the feature points, which together form the trajectory. Algorithm 1 shows the working of the proposed approach, given a smoothing radius.

Features points are found for each frame by using Shi-Tomasi Corner detection algorithm [4]. These feature points are used to obtain the corresponding matching features in the next consecutive frames using optical flow algorithm [3, 8]. Transformation matrix  $T_{original}^{(k)}$  is estimated from the feature points of frame  $k$  and frame  $k + 1$ , that encompasses the change between the two frames [11].  $T_{original}^{(k)}$  provides the *transform parameter* values as:

$$[dx^{(k)}, dy^{(k)}, da^{(k)}] = \left[ T_{original}^{(k)}[0, 2], T_{original}^{(k)}[1, 2], \tan^{-1} \left( \frac{T_{original}^{(k)}[1, 0]}{T_{original}^{(k)}[0, 0]} \right) \right] \quad (1)$$

---

**Algorithm 1:** Stabilisation using sliding window

---

**Input:** Shaky camera footage  
**Parameter :** Smoothing radius,  $r$   
**Output:** Stabilized video

- 1 Process each frame to obtain frame-to-frame transform  $\Delta_{original}$
- 2 Using  $\Delta_{original}$ , compute trajectories  $I_{original}$
- 3 **while**  $k < \text{Number of frames} - r$  **do**
- 4      $Sum = \sum_{i=k-r}^{i=k+r} I_{original}^i$
- 5      $WindowSize = 2 * r$
- 6      $I_{smooth}^k = Sum / WindowSize$
- 7 **end**
- 8 Compute  $\Delta_{smooth}$  to raise each point from its position in  $I_{original}$  to  $I_{smooth}$
- 9 Using  $\Delta_{smooth}$ , obtain and apply transformation matrices  $\Gamma$  on the sequence of frames
- 10 Resultant sequence of frames is the stabilized output

---

These values are stored as a list,

$$\Delta_{original} = \{tp^{(k)} | tp^{(k)} = (dx^{(k)}, dy^{(k)}, da^{(k)})\} \quad (2)$$

where,  $\Delta_{original}$  is then used to generate a single image trajectory,  $I_{original}$ . First, trajectory point is initialized to  $x=0, y=0, a=0$ . Each point thereafter is updated by adding to it a *transform parameter*. The final list contains the image trajectory,  $I_{original} = \{p^{(k)}\}$ . Camera shake is removed in the third step. For this, a new smooth trajectory,  $I_{smooth}$  is computed using a sliding average window algorithm. This requires a parameter, namely the constant smoothing radius  $r$ . The value of  $r$  is the number of frames on either side of the current frame used for the sliding window.

Figure 1 depicts that, as smoothing radius increases, stability tends to increase. But having arbitrarily high values of smoothing radius can be detrimental. As smoothing radius increases, the absolute values of the transformations applied to the frames increases leading to more data being lost as shown in Fig. 2. Data loss can be quantitatively defined as the ratio of black data-void areas in the stabilized frame to the original frame area. An optimal smoothing radius is thus desired.

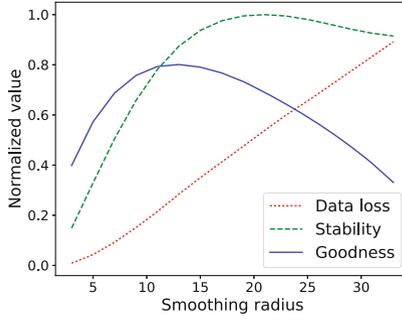
Let  $S_V(r)$  and  $D_V(r)$  represent the stability and data loss respectively in the video obtained by smoothing video  $V$  with a smoothing radius of  $r$ . Let  $s_V(r)$  and  $d_V(r)$  be the corresponding min-max normalized functions. As a high value of  $s_V(r)$  and a low value of  $d_V(r)$  is desirable, the *goodness* of a video is:

$$G(s_V(r), d_V(r)) = \sqrt{s_V(r) * (1 - d_V(r))} \quad (3)$$

and the optimal smoothing radius is then given by

$$r^* = \operatorname{argmax}_r(G(s_V(r), d_V(r))) \quad (4)$$

To reduce the time taken for calculation, following constraints are introduced.



**Fig. 1.** Effect of smoothing radii on goodness of a video, stability and data loss

1.  $0 < r \leq R_{max}$ : High radius causes high data loss and can be disregarded. The exact value of  $R_{max}$  is chosen empirically to be 40.
2.  $D_V(r) < 0.1$ : Fixing a range on just  $r$  is insufficient as the data loss function can ascend very quickly even within that range. Thus, to preclude that, a constraint on data loss is also introduced.

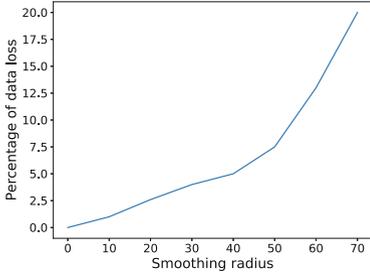
Having obtained the desired smooth trajectory, *transform parameter* values are required to transform every frame such that their old trajectory points are shifted to the ones in line with the smooth trajectory. For a particular frame, the distance between its  $I_{original}^{(k)}$  and  $I_{smooth}^{(k)}$  is calculated and added to the corresponding  $\Delta_{original}^{(k)}$ , to obtain a new *transform parameter*  $\delta_{smooth}^{(k)}$ . In effect, a point from a previous frame is translated to the current point of current frame and then translated from there to the smooth point of the current frame. These calculated values are brought together into a sequence by  $\Delta_{smooth} = \{\delta_{smooth}^{(k)}\}$ . Subsequently, the transformation matrix for the new smoothed trajectory and *transform parameter* values are computed using  $\Delta_{smooth}$ , and applied to produce a stable video output. Suppose smooth transform for the  $k^{th}$  frame is  $st^{(k)}$ , then the transformation matrix is given as,

$$T_{smooth}^{(k)} = \begin{bmatrix} \cos(da^{(k)}) & -\sin(da^{(k)}) & dx^{(k)} \\ \sin(da^{(k)}) & \cos(da^{(k)}) & dy^{(k)} \end{bmatrix} \quad (5)$$

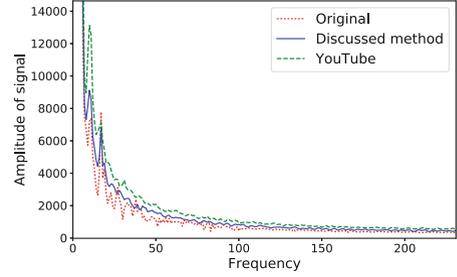
This affine transformation matrix is calculated for every frame to obtain the required sequence of transformations by  $\Gamma = \{T_{smooth}^{(k)}\}$ . Each transformation  $T_{smooth}^{(k)}$  is then applied on the corresponding frame  $k$ . The resultant sequence of frames is the stabilized video.

## 4 Results and Comparison

The proposed approach is tested on videos with and without objects of interest. To assess the performance of the proposed method, two performance criteria - stability and distortion - suggested by [6] are used to compare the proposed



**Fig. 2.** Effect of varying smoothing radius on data loss



**Fig. 3.** Frequency domain representation of the X coordinate signal

method with the YouTube stabilizer [2], and non-stabilized videos. Distortion between two continuous frames is the sum of pixel-by-pixel difference in intensity. Overall distortion of a video is the average of the pairwise distortion between continuous frames over the length of the video. A lower average implies a less distorted video. Table 1 compares the result of the proposed method with the original video and YouTube stabilized video. In most cases distortion is relatively higher than YouTube using the proposed method.

**Table 1.** Results of Distortion measure and Stability measure

Video	Distortion			Stability		
	Non-stabilized	Proposed method	YouTube	Non-stabilized	Proposed method	YouTube
1	0.4219	0.4179	0.4182	95.32	95.61	95.93
2	0.4003	0.3988	0.3729	95.57	95.57	95.68
3	0.4322	0.4206	0.3890	91.76	93.74	93.59

Stability approximates how stable the video appears to a viewer. Quantitatively, a higher fraction of energy present in the low frequency region of the Fourier transform of the estimated motion implies higher stability. Figure 3 shows the frequency domain representation of the X-coordinate motion. Over a dataset of 11 videos, it is observed that the proposed method shows an average of 3.92% improvement in stability over the original video, while the YouTube stabilizer shows a 4.28% improvement. The proposed method outperforms the YouTube stabilizer in three out of eleven videos. The results of this metric on three videos are shown in Table 1. The stability of the proposed method is usually comparable with that of YouTube, and in a few cases outperforms it. The proposed method does not handle sudden jerks well. Its performance is however unaffected by the presence or absence of an object in focus, thus making it applicable to a large class of videos.

## 5 Conclusions and Future Work

In this paper an algorithm for video stabilization has been proposed. This was accomplished by obtaining a global 2-D motion estimate for the optical flow of the video in both the  $X$  and  $Y$  directions. The algorithm is simpler than most current implementations, and provides comparable accuracy with YouTube's stabilizer. It also takes into account the image degradation to maintain optimal video quality. Taking into consideration multiple paths to estimate the motion [6] and motion inpainting [9] for image correction can improve the performance of this implementation. Future work can also include adaptively varying the smoothing radius over the video.

## References

1. Battiato, S., Gallo, G., Puglisi, G., Scellato, S.: Sift features tracking for video stabilization. In: 14th International Conference on Image Analysis and Processing, ICIAP 2007, pp. 825–830. IEEE (2007)
2. Grundmann, M., Kwatra, V., Essa, I.: Auto-directed video stabilization with robust 11 optimal camera paths. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 225–232. IEEE (2011)
3. Horn, B.K., Schunck, B.G.: Determining optical flow. *Artif. Intell.* **17**(1–3), 185–203 (1981)
4. Jianbo, S., Carlo, T.: Good features to track. In: Proceedings of 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1994, pp. 593–600. IEEE (1994)
5. Lee, K.Y., Chuang, Y.Y., Chen, B.Y., Ouhyoung, M.: Video stabilization using robust feature trajectories. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1397–1404. IEEE (2009)
6. Liu, S., Yuan, L., Tan, P., Sun, J.: Bundled camera paths for video stabilization. *ACM Trans. Graph. (TOG)* **32**(4), 78 (2013)
7. Liu, S., Yuan, L., Tan, P., Sun, J.: Steadyflow: spatially smooth optical flow for video stabilization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4209–4216 (2014)
8. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. In: Proceedings DARPA Image Understanding Workshop, pp. 121–130 (1981)
9. Matsushita, Y., Ofek, E., Ge, W., Tang, X., Shum, H.Y.: Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(7), 1150–1163 (2006)
10. Nghia, H.: Simple video stabilisation using opencv (2014), <http://nghiaho.com/?p=2093>. Accessed 17 Mar 2017
11. Nghia, H.: Understanding opencv cv::estimaterrigidtransform (2015), <http://nghiaho.com/?p=2208>. Accessed 15 Aug 2017
12. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571. IEEE (2011)