Stabilization of Jittery Videos Using Feature Point Matching Technique

S. Kulkarni[1], D. Bormane[2] and S. Nalbalwar[3]

[1] Research Scholar, JNTU Kakinada, India.
[2]Principal, JSPM's Narhe Technical Campus, Pune,
[3]Professor & Head (E&Tc), Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, India.

Abstract: Video capturing by non-professionals will lead to unanticipated effects. such as image distortion, image blurring etc. Many researchers focused on these drawbacks to enhance the quality of videos. In this paper an algorithm based on s-R-t transform is proposed to stabilize jittery videos .A stable output video is obtained without the effect of jitter which is caused due to shaking of handheld camera during video recording. In this technique firstly, salient points from each frame from the input video are identified using FAST algorithm. Camera motion is corrected by affine transform and motion compensation is performed by s-R-t transform which gives stabilized video. Optimization includes the quality of the video stabilization. This method has shown good result in terms of stabilization and it discarded distortion from the output videos recorded in different circumstances.

Keywords: video stabilization, point feature matching, Affine Transform, FAST algorithm, s-R-t Transform, image quality measurement

1. Introduction

Recently, the market of handheld camera has grown rapidly. However, video capturing by non-professionals or armatures will lead to unanticipated effects, such as image distortion, image blurring etc. When the shooting camera is mounted on the moving vehicle, then due to jerks, vibrations of the vehicle jittery video is produced. Hence, many researchers are working on such problems to enhance the quality of recorded l videos. Manufacturers have provided hardware and software support to the camera for enhancement, but still there are draw backs to overcome when multiple objects are moving in the background [1].

Digital video stabilization is the process of removing unwanted movement from a video stream. It is different from mechanical and optical stabilization. Mechanical stabilization physically dampens out vibration or unintended movement with gyroscopes. Optical stabilization modifies each input frame to maintain a steady image after it is converted to digital form [2].

Generally the processes of stabilization goes through three phases namely 1) Motion estimation 2) Motion smoothing 3) Motion compensation. The purpose of first phase is to estimate the motion of the moving objects in the frames. In the second phase parameters of estimated motion are sent to motion smoothening, where it removes the high-frequency distortion and calculates the global transformation.[4] Next, warping will be done for motion compensation to get stabilized video. These three-step frameworks are the essential steps in most of the video stabilization algorithms.

2. Literature Survey

Initial step in the video stabilization is selection of feature points or points of interest. These feature points are corners. In the late 1970's, Labeeb Mohsin Abdullah et.al.had proposed an algorithm to stabilize the video sequence by using Harris corner detection technique [9]. Harris Corner Detection is one of the fastest algorithms to find corner values. Edward Rosten et.al proposed faster and better machine learning approach to corner detection [4]. This algorithm also used principle of identifying Interest points in an image. Yue Wang et.al. had proposed Real-Time Video Stabilization for Unmanned Aerial Vehicles [11]. Elmar Mair et.al. had proposed technique of Adaptive and Generic Corner Detection based on the Accelerated Segment Test [3]. C. Harris and M.J. Stephens proposed combined corner and edge detection to deal with image regions containing texture and isolated features.

Their approach is based on the local auto-correlation function. Mohammed A. Alharbi, had proposed Fast Video Stabilization Algorithm [13] in which affine motion model is utilized to determine the parameters of translation and

B. Iyer, S. Nalbalwar and R. Pawade (Eds.)
ICCASP/ICMMD-2016. Advances in Intelligent Systems Research.
Vol. 137, Pp. 728-737.

rotation between images. The determined affine transformation is then used to compensate for the abrupt temporal discontinuities of input video sequence.

3. Methodology

The Flow chart of video stabilization algorithm is shown in Fig.1. The total process of video stabilization has been performed in several steps. These steps are elaborated with simulation results.

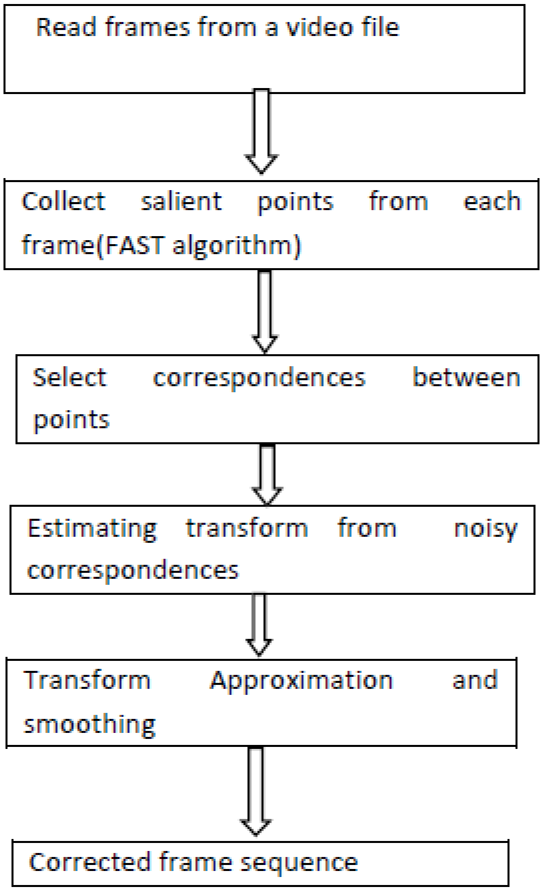
Read frames from a video file
Collect salient points from each frame(FAST algorithm)
Select correspondences between points
Estimating transform from noisy correspondences
Transform Approximation and smoothing
Corrected frame sequence

Fig.1. Flow chart of the of video stabilization algorithm

3.1 Read Frames from Video Sequence

In this step, first two frames of a video sequence are read. These frames are read as intensity images since color is not necessary for the stabilization algorithm and also, by using grayscale images speed also increases. The RGB to gray scale conversion is performed by taking weighted sum of the *R*, *G*, and *B* components given by following equation(1)

$$Y=0.2989 * R + 0.5870 * G + 0.1140 * B \qquad \text{-------}(1)$$

Fig.2shows both frames side by side. The pixel wise intensity of images has been separated from color. There is a large vertical and horizontal offset between the two frames because of movement of camera.

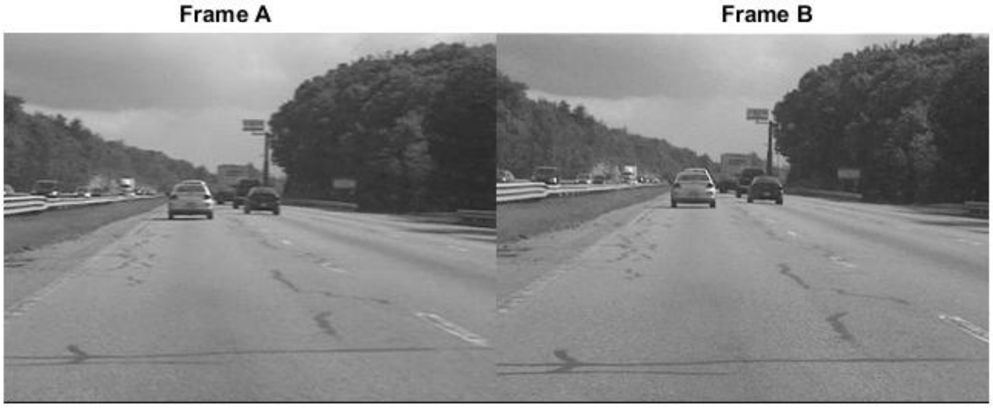

Fig. 2. Reading the two frames from a video sequence

After reading two frames, Red-Cyan color composite is produced to find pixel wise difference between them as shown in fig. 3.

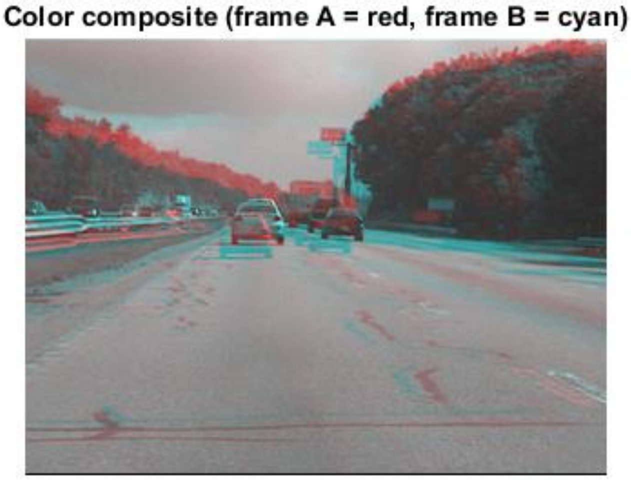

Fig. 3. Red-Cyan color composite

3.2 Collection of Salient Points (feature points) from each frame

Our aim is to find the transformation which will correct the distortion between the two frames, for this we have to collect feature points. Feature points are the points of interest such as corners, signs on the road, tops of trees, etc. All these points are having localized information [5]. For this purpose, there are several corner detection algorithms like: Moravec corner detection algorithm, Harris & Stephens's corner detection algorithm, SUSAN corner detector, etc. In our technique we use FAST method for detection of feature points. FAST (Fast Accelerated Segmented Test) is an algorithm for detection of feature points which gives high measured quality video with less computational time.

The detected points from both frames are shown in the fig.4; features such as corners of the cars, points along the tree line, corners of the large road sign etc. are covered in both frames A and B.

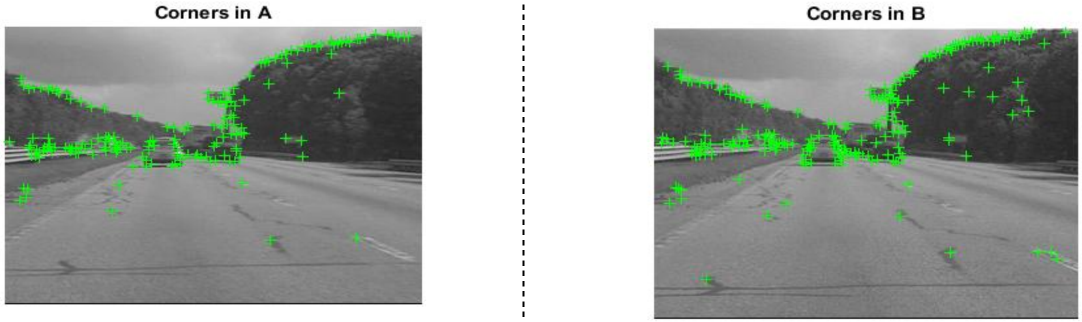

Fig. 4(a) corners in frame A

Fig. 4(b) Corners in frame B

Fig.4. Collection of salient points from both frames (a) corners in frame (b) Corners in frame B

3.3 Select Correspondences between Points

In this step, correspondences between the points derived in sec.3.2 should be established. For each point, we extract a Fast Retina Key point (FREAK) descriptor centered on it [7]. The matching cost is the sum of squared differences (SSD) between their respective image regions.
Let I(x, y) and I(x + u, y + v) are two images, then the SSD defined as

$$E(x,y) = \sum_{(x,y)\in w}[\mathrm{I}(\mathrm{x}+\mathrm{u},\mathrm{y}+\mathrm{v}) - \mathrm{I}(\mathrm{x},\mathrm{y})]^2 \quad \text{---(2)}$$

Taylor series expansion of I:

$$\mathrm{I}(\mathrm{x}+\mathrm{u},\mathrm{y}+\mathrm{v}) = \mathrm{I}(\mathrm{x},\mathrm{y}) + \frac{\partial \mathrm{I}}{\partial \mathrm{x}}\mathrm{u} + \frac{\partial \mathrm{I}}{\partial \mathrm{x}}\mathrm{v} + \text{higher order terms}$$

If motion (u, v) is small, then the first order approximation holds good.

$$\mathrm{I}(\mathrm{x}+\mathrm{u},\mathrm{y}+\mathrm{v}) \approx \mathrm{I}(\mathrm{x},\mathrm{y}) + \frac{\partial \mathrm{I}}{\partial \mathrm{x}}\mathrm{u} + \frac{\partial \mathrm{I}}{\partial \mathrm{x}}\mathrm{v}$$

$$\approx \mathrm{I}\,(\mathrm{x},\mathrm{y}) + [\mathrm{I}_{\mathrm{x}},\mathrm{I}_{\mathrm{y}}]\binom{u}{v} \quad - \quad \text{--(3)}$$

Where $\mathrm{I}_{\mathrm{x}} = \frac{\partial \mathrm{I}}{\partial \mathrm{x}}$
Plugging (2) into (1)

$$E(x,y) \approx \sum_{(x,y)\in w}[\mathrm{I}\,(\mathrm{x},\mathrm{y}) + [\mathrm{Ix},\mathrm{Iy}]\binom{u}{v} - \mathrm{I}(\mathrm{x},\mathrm{y})]^2$$

Hence

$$E(x,y) \approx \sum_{(x,y)\in w}[\,[\mathrm{Ix},\mathrm{Iy}]\binom{u}{v}]^2$$

This can be rewritten as

$$E\,(x,y) \approx [u\ v]\left(\sum_{(x,y)\in W}\begin{matrix} {I_x}^2 & I_xI_y \\ I_xI_y & {I_y}^2 \end{matrix}\right)\begin{matrix} u \\ v \end{matrix} \quad \text{----(4)}$$

Where H= distortion between two consecutive frames given by

$$H = \sum_{(x,y)\in W} \begin{matrix} {I_x}^2 & I_x I_y \\ I_x I_y & {I_y}^2 \end{matrix}$$

In order to find the directions of H, we need to find the Eigen values and Eigen vectors.
The Eigen vectors of matrix H are vectors x that satisfy

$$\mathrm{H.x} = \lambda.\ \mathrm{X}$$

The scalar λ is the Eigen value corresponding to x. The Eigen values are found by solving

$$\det(H - \lambda I) = 0$$

In this case H = 2 × 2 matrix, so we have

$$\det \begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix}$$

The solution is

$$\lambda_{\pm} = \frac{1}{2}\left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2}\right]$$

From Eigen values we can find out Eigen vectors using following form

$$\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

Choose points where λ- is local maximum as feature points (interest points). The above procedure is applied to show the correspondences between frames A and B. Thus points in frame A and frame B are matched closely. Fig.5 shows the same color composite given above, in this feature points from frame A are shown with red colour, and the feature points from frame B in green colour. Yellow lines are drawn between points to show the correspondences. There is no unique constraint, so points from frame B can correspond to multiple points in frame A. Many of these correspondences are correct, but still there is significant number of outliers [8].

3.4 Estimating Transform from Noisy Correspondences

Many of the point correspondences obtained in the previous step have limited accuracy. To rectify this problem, Random Sample Consensus (RANSAC) algorithm is used which is implemented in the Geometric Transform function in MatLab [10].

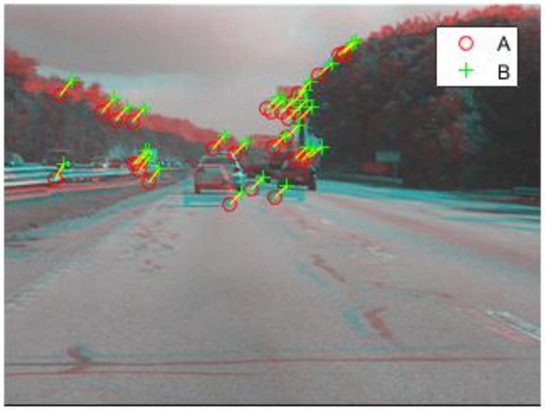

Fig.5 . Correspondences between Points

Using the Random Sample Consensus (RANSAC) algorithm, a robust estimate of transformation between Frame A and Frame B can be derived. RANSAC algorithm works to find effective inlier correspondences and afterward it derives the affine transformation to map the inliers in Frame A to Frame B. This transformation is only capable to alter the image plane.

It is observed that the inlier correspondences in the image background are not aligned with foreground. The reason behind this is the background features are far enough those act as if they were on an infinitely distant plane [11]. We can assume that background plane is static and will not change dramatically between the first and second frame. Affine transform captures the motion of the camera and thus correcting process will stabilize the video. Furthermore, as long as the motion of the camera between frame A and frame B is minimum or the time of sampling of video is high enough, this condition is maintained. The RANSAC algorithm is repeated multiple times and at each run the cost of the result is calculated by projecting frame B onto frame A via Sum of Absolute Differences between the two image frames [12].

SAD measures the distortion between two frames by evaluating the similarity between image blocks. Equation (5) defines the SAD between elements in two image blocks.

$$\mathrm{SAD} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left| c_{i,j} - r_{i,j} \right| \qquad \text{-----(5)}$$

Where ri,j represent elements in the first frame and ci,j are the elements in the second frame.

On the basis of SAD values, affine transform is obtained which makes the inliers from the first set of points to match with the inliers from the second set. This affine transform is 3-by-3 matrix and it can be expressed in the form of a matrix multiplication (linear transformation) followed by a vector addition (translation). Affine Transformation represents a relation between two images. The usual way to represent an Affine Transform is by using a 2×3 matrix.

$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}_{2\times 2} \quad B = \begin{bmatrix} b_{00} \\ b_{10} \end{bmatrix}_{2\times 1}$$

$$M = [A \quad B] = \begin{bmatrix} a_{00} & a_{01} & b_{00} \\ a_{10} & a_{11} & b_{10} \end{bmatrix}_{2\times 3}$$

Considering that we want to transform a 2D vector X=$\begin{bmatrix} x \\ y \end{bmatrix}$by using A and B,

We can do it equivalently with:

$$T = A \cdot \begin{bmatrix} x \\ y \end{bmatrix} + B$$

Or

$$T = M \cdot [x, y, 1]^{T}$$

$$T = \begin{bmatrix} a_{00}x + a_{01}y + b_{00} \\ a_{10}x + a_{11}y + b_{10} \end{bmatrix} \qquad \text{----(6)}$$

Affine Transformation gives information about a) Rotations b) Translations c) Scale operations which contains total six parameters as given by equation (6).

Affine transform can only alter the imaging plane. Fig. 6 shows a color composite showing frame A overlaid with the re-projected frame B, along with the re-projected point correspondences. It is clear from this figure, inliers correspondences gets exactly coincident which gives more favorable results. The cores of the images are both well aligned, such that the red-cyan color composite becomes almost purely black-and-white in that region.

3.5 Transform Approximation and Smoothing

We could use all the six parameters of the affine transform but for numerical simplicity and stability, we choose to re-fit the matrix as a simple scale-rotation-translation transform (s-R-t Transform).

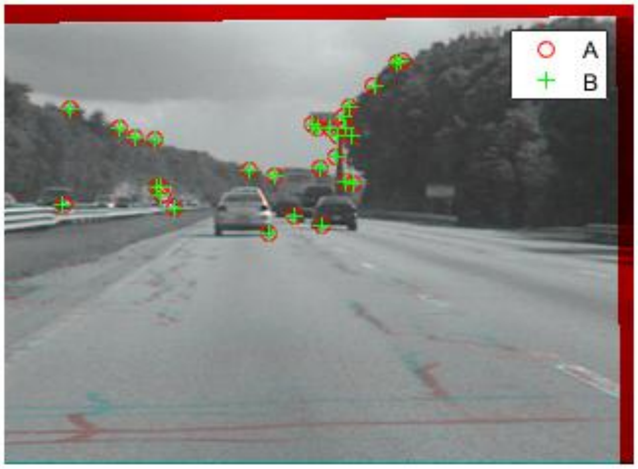

Fig. 6. Color composite showing frame A overlaid with the reprojected frame B

This transform has only four free parameters which are one scale factor, one angle, and two translations[13]. Construction of s-R-t transform is performed as follows:

- Extract scale and rotation part of sub- matrix from affine transform of matrix.
- Calculate theta from mean of two possible arctangents.
- Compute scale from mean of two stable mean calculations
- Translation will remain same.

This new transform matrix is of the form:

$$\begin{bmatrix} s * cos(ang) & s * -sin(ang) & 0 \\ s * sin(ang) & s * cos(ang) & 0 \\ t\_xt & t\_yt & 1 \end{bmatrix}$$

Finally by making use of s-R-t transform output video frame is shown in Figure 7.

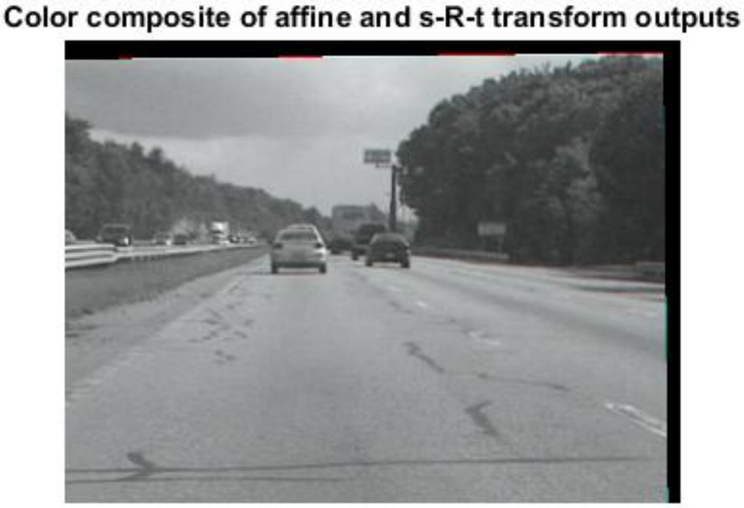

Fig.7. output video frame obtained by s-R-t Transform

3.6 Running full video through MATLAB video player

The last step of video stabilization algorithm is to run the above procedure in a loop for all frames in a video sequence. The process of estimating the transform between two images has been performed in the MATLAB® function **cvexEstStabilizationTform.** This function also converts a general affine transform into a scale-rotation-translation transform(s-R-t Transform). The last two frames of the smoothed video is shown in Video Player as a red-cyan composite.

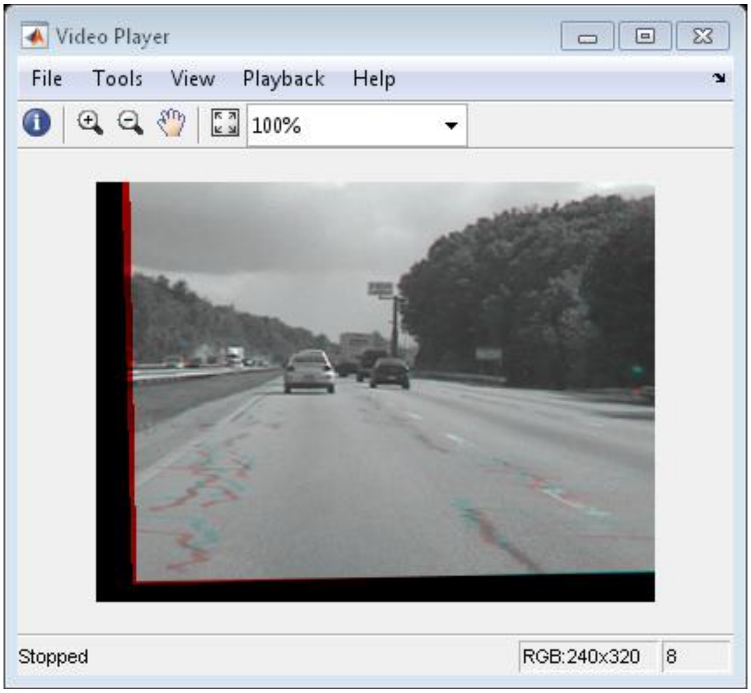

Fig.8. Last two frames of the smoothed video

4. Simulation Results

In this experimentation two shaky videos are considered. Video 1(shaky _car) has the following specifications: time duration 4 seconds, width x height of 320x240, frame rate of 30 frames/second and size of 1.2 MB. Video 2(shaky _fly) has the following specifications: time duration of 5 seconds, width x height of 480x288, frame rate of 30 frames/second and size of 2.07 MB. FAST detection method is applied for feature point detection. The stabilization technique is implemented using the MATLAB platform.

The improvement in the quality of the stabilized videos is evaluated on the basis of mean of the raw video frames and mean of the stabilized frames. These mean values are shown side-by-side below in fig.9 for Video-1(shaky_car). The left image shows the mean of the raw input frames and right image shows the mean of stabilized output frames. This proves that there was a great deal of distortion in the original video. The mean of the corrected frames on the right, shows the image with almost no distortion. The difference between video frames is mainly caused by the camera movement so the stabilization algorithm is efficient and the mean of stabilized video frames has almost no distortion. Foreground details have been blurred because of the car's forward motion. Fig.10 shows difference between mean of unprocessed video frames and mean of the corrected frames for Video-2 (shaky_fly).

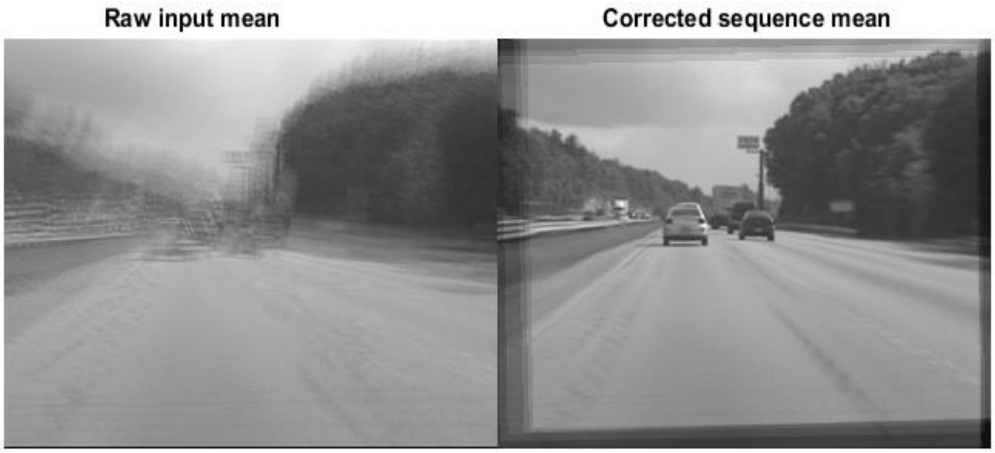

Fig. 9. Mean of the raw video frames and mean of the corrected frames for video-1(shaky_car)

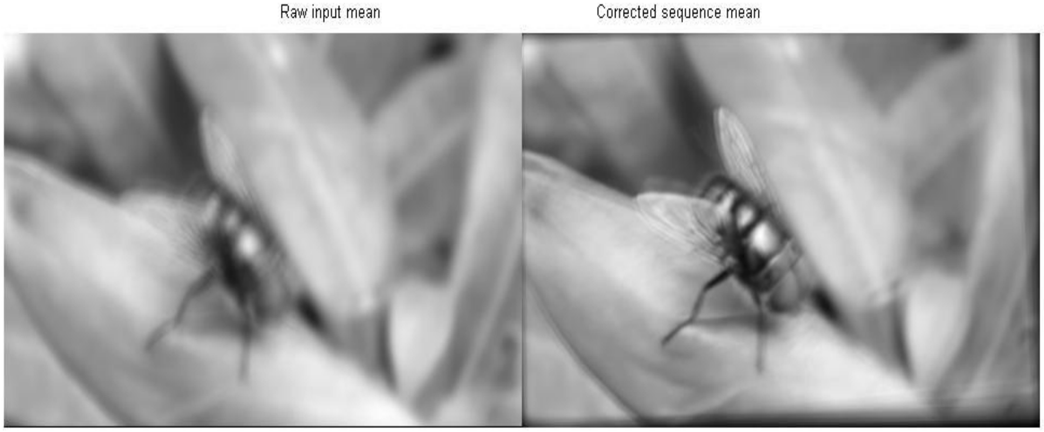

Fig.10. Mean of the raw video frames and mean of the corrected frames for Video-2(shaky_fly).

The quality and computational time are measured and compared for sample videos as shown in Table I.

Table- I Performance parameters

Sr.No	**Sample Video**	**Computational Time (s)**	**Quality Value**
1	Video-1(shaky_car)	8.81	22.20 %
2	Video-2(shaky_fly)	9.18	39.21%

5. Conclusion and Future scope

This technique provides logical and computationally efficient approach in terms of stabilizing high jitter videos suffered from distortion. The FAST detection algorithm presented in the work provides a fast and robust stabilization system and improves real-time performance.

Comparing stabilized and shaky video it gets confirmed that the processed videos highly satisfy the human perception. Results indicate a remarkable elimination of high jitter from shaky videos.

FAST detection technique is useful in enhancing the quality of low-grade video surveillance cameras. This video stabilization technique is particularly helpful in identifying people, license plates, etc. from low-quality video surveillance cameras.

In future, we can find better feature detector to overcome the consequences of extreme shaking of handheld camera in real time implementation for video stabilization.

References

[1] Aleksandra Shnayderman, Alexander Gusev, and Ahmet M. Eskicioglu :An SVD-Based Grayscale Image Quality Measure for Local and GlobalAssessment. IEEE 15(2), 2006.

[2] B.-Yu. Chen, K.-Yi. Lee, W.-T. Huang, J.-S. Lin.:Capturing Intention-based Full-Frame Video Stabilization.Computer Graphics Forum, Vol. 27, No. 7, p.1805 - p.1814, 2008.

[3] C. Harris and M.J. Stephens.:A combined corner and edge detector. Proc of Alvey Vision Conference, pp 147–152, 1988.

[4] C. Morimato and R. Chelleppa.: Evaluation of Image Stabilization algorithms. Proc. of IEEE int. Conf. on Acoustics, speech and signal processing, Vol-5,pages 2789-2792, 1995.

[5] Edward Rosten, Reid Porter and Tom Drummond:FASTER and better: A machine learning approach to corner detection. In IEEE Trans. Pattern Analysis and Machine Intelligence, 2010, vol 32, pp. 105-119

[6] Jie Xu, Hua-wen Chang, Shuo Yang, and Minghui Wang:Fast Feature-Based Video Stabilization without Accumulative Global Motion Estimation. IEEE 2012.

[7] Ken-Yi Lee Yung-Yu Chuang Bing-Yu Chen Ming Ouhyoung.:Video Stabilization using Robust Feature Trajectories. 2009 IEEE 12th International Conference on Computer Vision (ICCV).

[8] Keng-Yen Huang, Yi-Min Tsai, Chih-Chung Tsai, and Liang-Gee Chen.:Feature-based Video Stabilization for Vehicular Applications. 2010 IEEE 14th International Symposium on Consumer Electronics.

[9] Labeeb Mohsin Abdullah, Nooritawati Md Tahir & Mustaffa Samad .:Video Stabilization based on Point Feature Matching Technique. 2012 IEEE Control and System Graduate Research Colloquium (ICSGRC 2012).

[10] Xie Zheng, Cui Shaohui, Wang Gang, Li Jinlun.:Video Stabilization System Based on Speeded-up Robust Features. International Industrial Informatics

[11] Yang Zhang, Yuquan Leng, Xu He Member.:A Fast Video Stabilization Algorithm with Unexpected Motion Prediction Strategy.2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM) July 7-11, 2015. Busan, Korea.

[12] Z. Duric and A. Rosenfield.:stabilization of images sequences. Technical report CAR-TR-778, Center for Automation Research University of Maryland, College, Park, 1995.

[13] Thesis on Fast Video Stabilization Algorithms by Mohammed A. Alharbi, Captain, RSAF AFIT/GCS/ENG/06-02.