SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

AIVP ADVANCES IN IMAGE AND VIDEO PROCESSING

VOLUME 5, ISSUE 1
ISSN 2054-7412

RANSAC Algorithm for Matching Inlier Correspondences in Video Stabilization

S.M.Kulkarni[1], Dr.D.S.Bormane[2], Dr.S.L.Nalbalwar[3]
[1]*Research Scholar, JNTU Kakinada, India.*
[2]*Principal,JSPM's Narhe Technical Campus,,Savitribai Phule Pune University , Pune, India.*
[3]*Professor & Head(E&Tc) ,Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, India.*

ABSTRACT

In order to stabilize a video sequence we need to find a transformation which reduces the distortion between frames. To find this transformation feature points must be identified in consecutive frames. In order to get the correspondences between feature points Sum of Squared Differences (SSD) is adopted as matching cost between respective points but by this technique, many of the point correspondences are obtained and they have limited accuracy. To rectify this dilemma, Random Sample Consensus (RANSAC) algorithm is used which is implemented in the Geometric Transform function in Matlab. Utilizing the Random Sample Consensus (RANSAC) algorithm, a robust estimate of transformation between consecutive video frames could possibly be derived.

In this paper RANSAC algorithm can be used to find effective inlier correspondences and afterward it derives the affine transformation to map the inliers in consecutive video frames. This transformation is capable to improve the image plane .The RANSAC algorithm is repeated multiple times and at each run the cost of the end result is calculated via Sum of Absolute Differences between both image frames. SAD measures the distortion between two frames by evaluating the similarity between image blocks. On the cornerstone of SAD values, affine transform is obtained which makes the inliers from the initial set of points to match with the inliers from the following set. It is clear from simulation results, inliers correspondences gets exactly coincident which gives more favorable results. The cores of the images are generally well aligned. Thus by utilizing the Random Sample Consensus (RANSAC) algorithm, a robust estimate of transformation is obtained.

Key term: Video Stabilization, RANSAC Algorithm, Inlier and outliers correspondences, Affine Transform,s-R-t Transform

1 Introduction

Estimation of model parameters from an image is a critical computational problem in vision. The name robust estimation is given to estimations which are tolerant in the clear presence of outliers. The word 'Outlier' means a thing that lies outside the main body or group. In technical definition, if a data doesn't fit in with the 'true 'model defined by the 'true 'group of parameters considering some threshold value, it is going to be referred to as outlier. Robust estimation targets to find a set of inliers from the correspondences. RANdom SAmple Consensus abbreviated as RANSAC targets to estimate a mathematical model parameters based on some observations including outliners. This paradigm has been developed by M. A. Fischler and R. C. Bolles [1] with the goal of fitting a model to experimental data. Most of robust estimation techniques, gather the maximum amount of the data to acquire an

DOI: 10.14738/aivp.51.2692
Publication Date: 08th March, 2017
URL: http://dx.doi.org/10.14738/aivp.51.2692

initial solution, however in contrast, RANSAC uses the minimum feasible data set and when possible, tries to enlarge this set with consistent data.

RANSAC (RANdom SAmple Consensus) algorithm is widely used to detect a unique transformation by randomly sampling n-uplets[2]. For each one of these n-uplets, the corresponding transformation(s) is estimated and the adequacy of the transformation to the rest of the data is then tested. The transformation that maximizes this adequacy (yields the most effective consensus) is then kept. This process has two main advantages, first is robustness and secondly its speed. Nevertheless, its practical use requires the setting of several sensitive parameters. Several approaches have been proposed allowing the detection of multiple groups with RANSAC. These can be roughly categorized as follows.

First strategy is always to detect all groups simultaneously by fusing the various groups found by RANSAC [3]. A second strategy is always to sequentially detect groups by iteratively running RANSAC [4]. While the practical implementation of the simple idea is non-trivial, we will advocate the usage of the sequential approach and detail our algorithm for the detection of multiple groups.

2 Random Sample Consensus (RANSAC)

The RANSAC algorithm is basically made up of two steps which are repeated in a iterative process. First faltering step is Hypothesis and next step is Test.

Apriori information, that is used in the process of fitting the model includes:

1. Minimum quantity of points (observations) required to match the model that is also referred to as minimal sample sets (MSS)
2. Minimum quantity of iterations
3. The threshold (T) that splits the inliers from outliers along the way of hypothetical model testing
4. size of data set, which completes the procedure of the iteration.

The hypothesis step relates to the concept of the minimal sample set (MSS). The initial minimal sample set is randomly selected from the input dataset. The model parameters are computed using only the elements of the MSS. The MSS is the total amount of data (observations) needed to compute the parameters of the model .The first phase starts with selecting an essential and minimum quantity of observations of the data set. Based on these selected observations, the output model (hypothetical) is estimated [5]. All the remaining data is tested to match the hypothetical model. If one or both of selected points are burdened with outliers, then hypothetical model will not fit the rest of data. Therefore the algorithm will skip this model and randomly pick another set of points for another hypothetical model.

In the testing step, RANSAC iteratively checks which observations of entire dataset are consistent with the hypothetical model[6]. This calls for determining the threshold 'T' which specifies the maximum distance from an examination point to a hypothetical model. When it fits the criterion the point is treated as just hypothetical inlier. The estimated model is correct when it has sufficient amount of points which were classified as correct observations (inliers). The most effective group of observations which is selected from the whole dataset is known as the consensus set (CS)[7].

Consider $D=\{d_1,\ldots,d_N\}$could be the input data set through which you want to estimate a model. If $\theta(d_1\ldots, d_N)$ could be the parameter vector estimated by $\{d_1,\ldots, d_h\}$ and h is greater compared to the minimum amount of elements needed to estimate the model, the model space $\mathcal{M}$ of the parameter vector θ is going to be as follows [8].

$$\mathcal{M}(\theta) = \{d \in \mathcal{R}^d : \mathcal{F}_{\mathcal{M}}(d;\theta) = 0\} \qquad (1)$$

Where θ is a parameter vector and $\mathcal{FM}$ is a smooth function. Assuming condition that $\mathcal{FM}$ is equal to zero, it will contain all points fitting the model. The length from the datum *d* to the model $\mathcal{M}$ represents the error [9]. Datumn with an error value greater than certain threshold value are not consistent with the model.

$$e_M(d,\theta) = \min_{d' \in \mathcal{M}(\theta)} dist(d,d') \qquad (2)$$

So the consensus set CS is defined as:

$$S(\theta) = \{d \in D:\ e_M(d;\theta) \leq \delta\} \qquad (3)$$

Where δ is a threshold value which is often determined automatically based on some hypothesis.

The RANSAC Algorithm could be summarized while the follows:

1. In the first step of RANSAC paradigm, Minimal Sample Sets (MSSs) are chosen from from sample of random data points(S), from which the model is investigated.
2. In second step the paradigm introduces Consensus Set (CS) of the sample. This Consensus Set (CS) is termed as S_i. Actually Si offers the inliers of S. Inliers selection is based on a distance threshold value T_1 of the model.
3. In third step of the paradigm another threshold value T_2 is introduced. If there are more inliers than the value of T_2, all of the points in Si should be used to re-approximate the model.
4. If there is less number of inliers than the value of T_2, a brand new subset should be selected and the previous steps will be repeated.

The procedure will continue for N trials and finally the biggest consensus set Si is chosen. All of the points in this set are again used to re-approximate the model.Fig.1 demonstrates different steps of RANSAC.

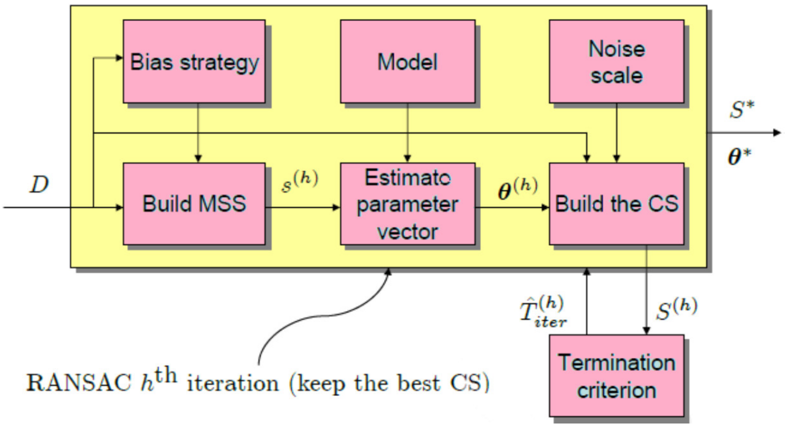


Figure.1: Fundamentals of RANSAC Iteration

2.1 Amount of Iterations to estimate the true model

Sampling of Minimal sample set(S) can result in precise approximation of model parameters, which has the probability *q*, which means probability of sampling a MSS containing one or more outliner will be 1-*q* [10]. If we Sample *h* MSSs then the probability that all of them contain outliers is

$(1-q)^h$ .The requirement is to decide number of iterations large enough to reduce the probability $(1-q)^h$ less than the usual threshold value, in a way that

$$h \geq \left\lceil \frac{\log \varepsilon}{\log 1 - q} \right\rceil \tag{4}$$

Where *h is* quantity of minimum sample spaces (MSS)

ε - Likelihood of incorrect identification of the model,

q - Likelihood of constructing a MSS containing just inliers

Therefore the threshold value for iterations may be set to:

$$\widehat{T_{iter}} = \left\lceil \frac{\log \varepsilon}{\log 1 - q} \right\rceil \tag{5}$$

If the probabilities of selecting each elements of the dataset are equal, then probability of constructing a MSS containing just inliers is given by equation (6).

$$q = \frac{\binom{N_I}{k}}{\binom{N}{k}} = \frac{N_I!\,(N-k)!}{N!\,(N_I-k)!} \tag{6}$$

Where *q*- probability of constructing a MSS containing just inliers.

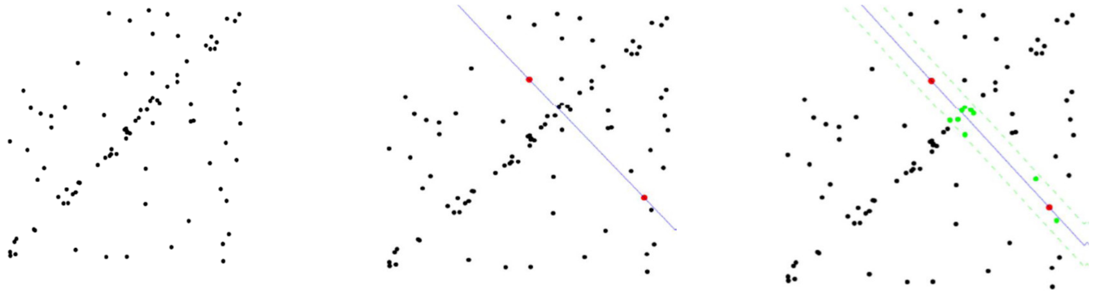
N_i - quantity of points that belong to the consensus set (total quantity of inliers)

N - Total quantity of points

k - Minimal quantity of data needed to define the model.

2.2 Example of Using RANSAC

In this section we present a good example of using RANSAC paradigm to find a line fitting for many of points as in Fig.2 (a). The minimum amount of required points to form a line is two, so in first step, algorithm selects two random points as MSS and estimates the line passing through them as shown in Fig.2 (b). Then error function is calculated and points in the certain threshold value are selected. This loop will be repeated for many iteration until the model containing the maximum amount of inliers is available as shown in fig.2(c).



(a) (b) (c)

Figure.2 Using RANSAC to find a Line Passing a number of Points

2.3 Maximum Likelihood Estimation Sample Consensus (MLESAC) and M-estimator Sample Consensus (MSAC)

If in RANSAC algorithm the threshold value determining inliers is considered very high, the robust estimation will soon be poor. Then for improving the quality of the consensus set **M-estimator Sample Consensus (MSAC) is** used [11].

Considering two images corrupted by zero mean Gaussian noise with standard deviation then probability density function of data will soon be as follows. [12]

$$P_r(D|M) = \prod_{i=1\ldots n} \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n e^{\left(\sum_{j=1,2}\left(\underline{x_i^j}-x_i^j\right)^2+\left(\underline{y_i^j}-y_i^j\right)^2\right)/(2\underline{\sigma}^2)} \tag{7}$$

Where *D* represents the matches set
n is number of correspondences
M could be the transformation between both images.
The following equation represents the negative logarithm of likelihood for all correspondences [13].

$$P_r(D|M) = \prod_{i=1\ldots n} \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n e^{\left(\sum_{j=1,2}\left(\underline{x_i^j}-x_i^j\right)^2+\left(\underline{y_i^j}-y_i^j\right)^2\right)/(2\underline{\sigma}^2)} \tag{8}$$

Defining the function *C* as a cost function, RANSAC algorithm finds the minimum value by utilizing following equation.

$$C = \sum_i \rho(e_i^2) \tag{9}$$

Where ρ is robust error term which is given by

$$\rho(e^2) = \begin{cases} 0 & e^2 < T^2 \\ constant & e^2 \geq T^2 \end{cases} \tag{10}$$

So by increasing the value of T^2 there will be more solutions with same value of C which results in poor estimation. Choosing *T* large enough, most of the matches will be inliers. We are able to minimize a new cost function instead of minimizing *C*.

$$\rho_2(e^2) = \begin{cases} e^2 & e^2 < T^2 \\ T^2 & e^2 \geq T^2 \end{cases} \tag{11}$$

It can be seen that outliers are given a fixed penalty nevertheless now inliers are scored how well they fit the data. The implementation with this new method is named as MSAC (*m-estimator sample consensus*).

3 Simulation Results

The procedure of matching inliers correspondences and video stabilization is explained in following steps with simulation results. In this experimentation jittery video (shaky_fly) is considered. It's specifications are: time duration 6 seconds, width x height of 480x288, frame rate of 30 frames/second and size of 2.07 MB.

3.1 Read Frames from Video Sequence

In this task, first two frames of a video sequence are read. These frames are read as intensity images since color is not needed for the stabilization algorithm and also, by utilizing gray scale images speed also increases [14]. Fig.3 shows both frames side by side. The pixel wise intensity of images has been separated from color. There is a sizable vertical and horizontal offset between the two frames as a result of movement of camera.

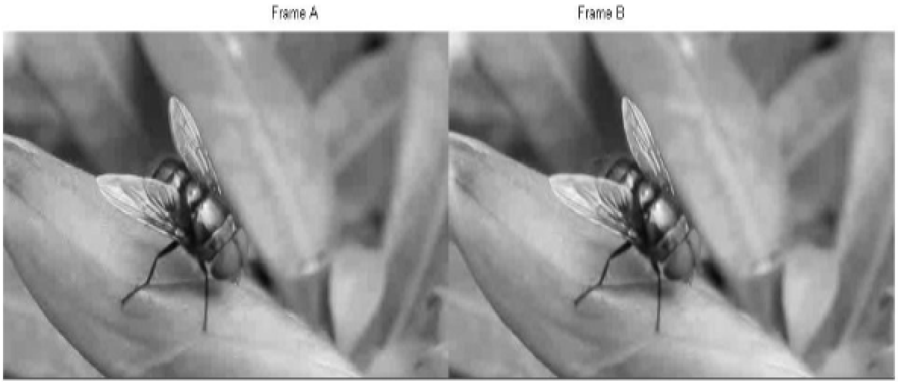


Figure 3: Reading the two frames from a video sequence

After reading two frames, Red-Cyan color composite is produced to distinguish pixel wise difference between them as shown in fig.4.

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

Our aim is to acquire the transformation which will correct the distortion between the two frames, because of this we have to gather feature points. Feature points in this video could be fins, legs of fly, etc. All these points are having localized information [15]. We use FAST method for detection of feature points.

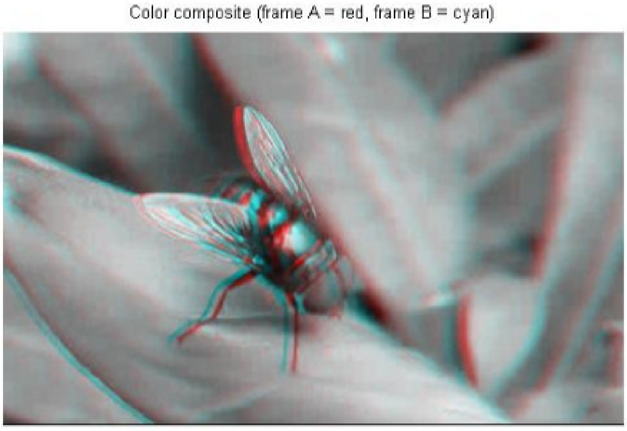


Figure. 4 Red-Cyan color composite

FAST (Fast Accelerated Segmented Test) can be an algorithm for detection of feature points which gives high measured quality video with less computational time [16].The detected points from both frames are shown in the fig.5, features such as, legs, fins of fly etc. are covered in both frames A and B.

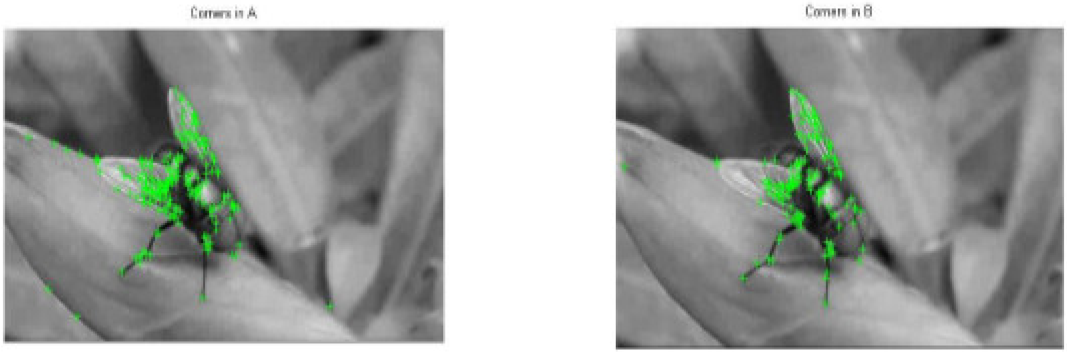


Figure. 5(a) corners in frame A

Figure. 5(b) Corners in frame B

Figure.5 Identification of salient points from both frames (a) corners in frame A (b) Corners in frame B

3.3 Select Correspondences between Points

In this task, correspondences between the points derived in sec.3.2 should be established. For each point, we extract a Fast Retina Key point (FREAK) descriptor devoted to it. The matching cost could be sum of squared differences (SSD) between their respective image regions. Fig.6 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 match multiple points in frame A. Several correspondences are correct, but there is significant quantity of outliers.

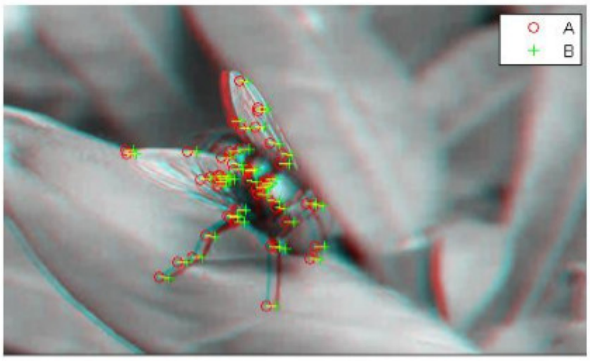


Figure.6 Correspondences between Points

3.4 Estimating Transform from Noisy Correspondences

Many of the point correspondences obtained in the last step have limited accuracy. To rectify this problem, Random Sample Consensus (RANSAC) algorithm discussed above can be utilized which is implemented in the Geometric Transform function in Matlab[13]. By utilizing the Random Sample Consensus (RANSAC) algorithm, a strong estimate of transformation between Frame A and Frame B could be derived.

RANSAC algorithm works to locate effective inlier correspondences and afterward it derives the affine transformation to map the inliers in Frame A to Frame B. This transformation is capable to boost the image plane [17].

It is observed that the inlier correspondences in the image background are not aligned with foreground. The reason for this is background features are far enough those act as if they were on an infinitely distant plane. We believe that background plane is static and will not change dramatically between the first and second frame. Furthermore, as long as the motion of the camera between frame A and frame B is minimum or time of sampling of video is high enough, stabilization of video takes place. The RANSAC algorithm is repeated multiple times and at each run the cost of end result is calculated by projecting frame B onto frame A via Sum of Absolute Differences between the two image frames.

SAD measures the distortion between two frames by evaluating the similarity between image blocks. Equation (12) defines the SAD between elements in two image blocks.

$$\text{SAD} = \sum_{i=1}^{N} \sum_{j=1}^{N} |c_{i,j} - r_{i,j}| \tag{12}$$

Where $c_{i,j}$ represent elements in the very first frame and $r_{i,j}$ are the elements next frame.

On the cornerstone of SAD values, affine transform is obtained helping to make the inliers from the very first set of points to suit with the inliers from the next set. This affine transform is 3-by-3 matrix and it is expressed in the shape of a matrix multiplication (linear transformation) followed with a vector addition (translation)

Affine Transformation gives information about a) Rotations b) Translations c) Scale operations which contain total six parameters Affine transform can only alter the imaging plane. Fig.7 shows a same composite showing frame A overlaid with the re-projected frame B. It is clear from this figure, inliers correspondences gets exactly coincident which provides 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 for that region. Affine transform captures the motion of the camera and thus correcting process will stabilize the video.

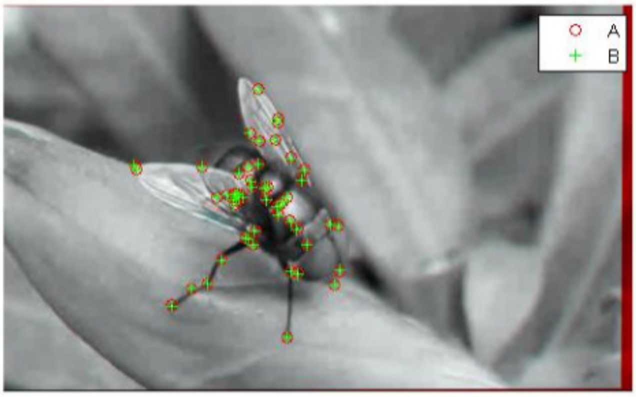


Figure. 7: RANSAC Algorithm and Affine transform for matching inlier correspondences

3.5 Transform Approximation and Smoothing

We could l use all six parameters of the affine transform but from numerical simplicity and stability, we choose to re-fit the matrix as an easy scale-rotation-translation transform (s-R-t Transform). This transform has only four free parameters which are one scale factor, one angle, and two translations.

s-R-t transform is constructed the following:

i) Extract scale and rotation section of sub- matrix from affine transform of matrix

ii) Calculate theta from mean of two possible arctangents iii)Compute scale from mean of two stable mean calculations

iv) 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 Fig. 8.

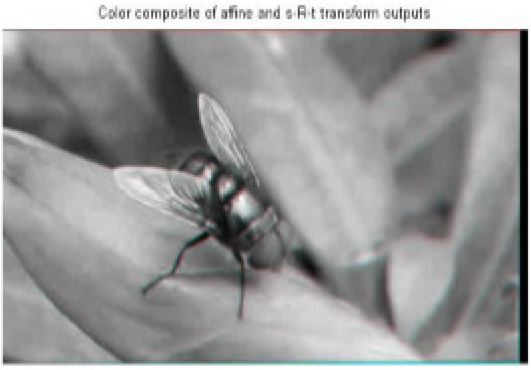


Figure. 8 output video frame obtained by s-R-t Transform

3.6 Running full video through MATLAB video player

The final step of video stabilization algorithm is to operate above mentioned procedure in a cycle for all l frames in a video sequence. The strategy of estimating the transform between two images has been performed in the MATLAB® function **cvexEstStabilizationTform.** This function also converts affine transform in to a scale-rotation-translation transform(s-R-t Transform). Fig.9 shows the last two frames of the smoothed video in Matlab Video Player as a red-cyan composite.

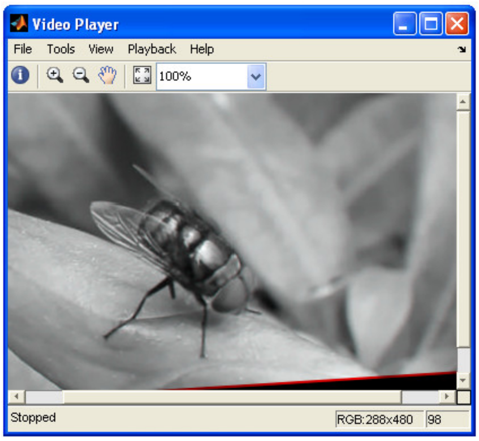


Figure.9 Last two frames of the smoothed video

The improvement in the quality of the stabilized videos is evaluated on the basis of Mean of unprocessed and stabilized video frames. We computed the mean of the raw video frames and mean of the stabilized frames. These mean values are shown side-by-side below in fig.10.

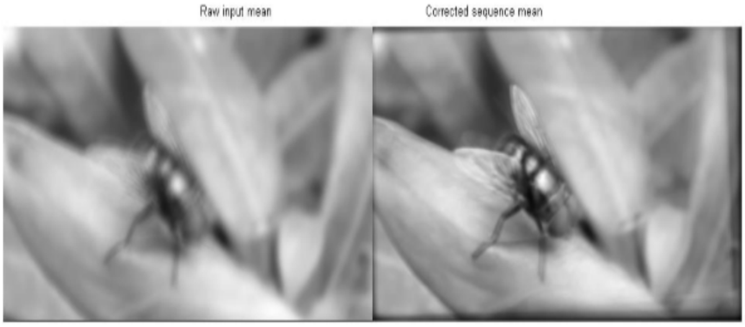


Figure. 10 mean of the raw video frames and mean of the corrected frames for video-1(shaky_fly)

The left image shows the mean of the unprocessed shaky video frames and right image shows the mean of stabilized output frames. This proves that there is a great deal of distortion in the first video. The mean of the corrected frames on the right, shows the image with minimal distortion. The difference between video frames is mainly due to the camera movement hence stabilization algorithm is efficient and the mean of stabilized video frames has minimal distortion.

4 Conclusion

This technique provides logical and computationally efficient approach with regards to stabilizing high jitter videos endured distortion. The RANSAC algorithm has the capacity to locate a correct solution and to remove outliers from estimation process. Therefore, the potency of this algorithm is placed on matching the correspondences between two frames.

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

Manner of video stabilization using point feature matching is advantageous in enhancing the grade of low-grade video surveillance cameras and and yes it is very useful 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] M. Fischler and R. Bolles. “Random sample consensus: a paradigm for model fitting with application to imageanalysis and automated cartography”, Commun. Assoc. Comp. Mach., vol. 24:381–95, 1981.

[2] R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In Proc. 2nd EuropeanConference on Computer Vision, LNCS 588, Santa Margherita Ligure, pages 579–587. Springer-Verlag,1992.

[3] P. J. Rousseeuw. Robust Regression and Outlier Detection. Wiley, New York, 1987.

[4] C. Tomasi and T. Kanade. “Shape and motion from image streams under orthography: A factorisation approach”, International Journal of Computer Vision, 9(2):137–154, 1992.

[5] P. H. S. Torr and D. W. Murray, “Outlier detection and motion segmentation”, In P. S. Schenker, editor, Sensor Fusion VI, pages 432–443. SPIE volume 2059, 1993. Boston.

[6] O. Chum and J. Matas. “Matching with PROSAC - progressive sample consensus”, In Proc. CVPR, pages 220–226,2005.

[7] J. Rabin, J. Delon, and Y. Gousseau. A statistical approachto the matching of local features. SIAM Journal on ImagingSciences, 2(3):931–958, 2009.

[8] P. H. S. Torr and A. Zisserman. MLESAC: a new robust estimator with application to estimating image geometry. Com-put. Vis. Image Underst., 78(1):138–156, 2000.

[9] W. Zhang and J. Kosecka. Generalized RANSAC Frame work for Relaxed Correspondence Problems. In Proceedings of 3DPVT’06.

[10] Z. Duric and A. Rosenfeld, "Stabilization of image sequences," Center forAutomation Research, University of Maryland, College Park, 1995.

[11] Matsushita Y., Ofek E., Tang X., and Shum H.Y, "Full-frame Video Stabilization with Motion Inpainting," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 7, pp. 1150 - 1163, July 2006.

[12] Torr P.H.S., and Zisserman A, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," Computer Vision and ImageUnderstanding, vol. 78, no. 1, p. 138–156, 2000.

[13] Puglisi G. , and Battiato S., "A Robust Image Alignment Algorithm for Video Stabilization Purposes," IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 10, pp. 1390 - 1400, 2011.

[14] Abdullah L.M., Tahir N. M., and Samad M, "Video Stabilization Based on Point Feature Matching Technique," in Control and System Graduate Research Colloquium (ICSGRC), 2012 IEEE, Shah Alam, Selangor, 2012.

[15] Choi S., Kim T., Yu W, "Robust video stabilization to outlier motion using adaptive RANSAC," in International Conference on Intelligent Robots and Systems, St. Louis, MO, 2009.

[16] I Sack H., Boykov Y., Energy-based Geometric Multi-Model Fitting. *International Journal of Computer Vision.* 97(2): 1: 23-147, 2012

[17] Janicka, J., Rapapinski J., M-split transformation of coordinates. *Survey Review.* Vol. 45, issue 331, pp. 269-274, 2013