ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Video Stabilization for Micro Air Vehicles

Karra Somasekhar Reddy[1], Lokesha.H[2], S.Akhila[3]

M.Tech, Digital Communication, E & C Department, BMS College of Engineering, Bangalore, India [1]

Senior Scientist, ALD Division, DSP Lab, CSIR-National Aerospace Laboratories, Bangalore, India [2]

Professor, E & C Department, BMS College of Engineering, Bangalore, India [3]

Abstract: In Today's world, Micro Air Vehicles (MAVs) equipped with digital cameras are increasing in number for real time applications. MAVs are widely used because of small, low weight, low cost and flexibility in the design. However, the video captured by MAVs will consist of unwanted translations and rotations due to atmospheric conditions. Hence, a video stabilization algorithm is needed to compensate the frames and then to avoid losing of any valuable information in video. This paper presents a real time video stabilization algorithm and mainly consists of three steps. Firstly, the key points from each frame are calculated by using Harris Corner Detection algorithm. Secondly, the descriptors between the two frames are used to calculate the affine transformation. Finally, this affine transformation is used to compensate the two successive frames in a video. This work mainly focuses on the digital image processing based video stabilization. The programming was primarily using MATLAB and later the algorithm was implemented on Raspberry Pi using Opencv-Python Programming.

Keywords: Micro Air Vehicles, Video stabilization, Harris Corner Detection, Affine transformation, Motion Compensation.

I. INTRODUCTION

Micro air vehicles (MAVs) are a class of Unmanned Aircraft Vehicles (UAVs) significantly smaller than typical UAVs. UAV is an aircraft without a human pilot on board. It is controlled either automatically by on board computers or by the remote control pilot on the ground or by another vehicle [5].

The MAVs provide video surveillance during military exercises, firefighting, and weather monitoring, policing and border control. However the video captured by MAVs consist of unwanted shakes and jitter due to low weight, inexpensive cameras that are accomplished on vibrating engine and weather conditions. Hence implementation of video stabilization algorithm for MAVs is a unique challenge.

Video stabilization is a process used to improve video quality by removing unwanted camera shakes and jitters. The removal of unwanted vibrations in a video sequence induced by camera motion is an essential part of video acquisition in industry, military and consumer applications.

The video stabilization can be achieved either by hardware or digital image processing approach. Hardware approach can be further divided as mechanical or optical stabilization. Mechanical stabilizer uses gyroscopic sensor to stabilize entire camera. Optical stabilization activates an optical system to adjust camera motion sensors [1].These techniques are not suitable for small camera modules embedded in mobile phones like devices due to lack of their compactness and also the associated cost. The image processing approach tries to smooth and compensate the undesired motions by means of digital video processing. Digital image stabilization (DIS) techniques, without resorting to any mechanical device (like gyros) or optical devices (like fluid prisms), are popular due to their low cost, fast response and compactness.

This paper is arranged as follows. The basic concept of general video stabilization algorithm is discussed in Section II. The proposed video stabilization algorithm in Section III and the experimental results and discussions are presented in Section IV. The paper concludes with Section V.

II. BASIC CONCEPT OF VIDEO STABILIZATION

In general, any digital video stabilization algorithm consists of three stages: motion estimation (ME), motion smoother (MS) and motion compensation (MC) as in Fig. 1 [2]. ME estimate the motion between the frames, and send the motion parameters to MS, which removes the unwanted camera motions. MC then computes the global transformation necessary to stabilize the current frame.

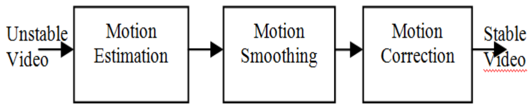


Fig. 1: General Block diagram of Video Stabilization

III. PROPOSED METHOD OF VIDEO STABILIZATION

The flowchart of video stabilization algorithm that was implemented on Raspberry Pi is depicted in Fig. 2. Firstly, the corners of frames are extracted by Harris and Stephens corner detection algorithm and matching points are determined. Next, the motion between two consecutive frames is estimated based on an affine transform model [3]. Finally, the frames are compensated based on

DOI 10.17148/IJARCCE.2015.44101

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

cumulative transform and form a stable video. The detailed are explained as follows.

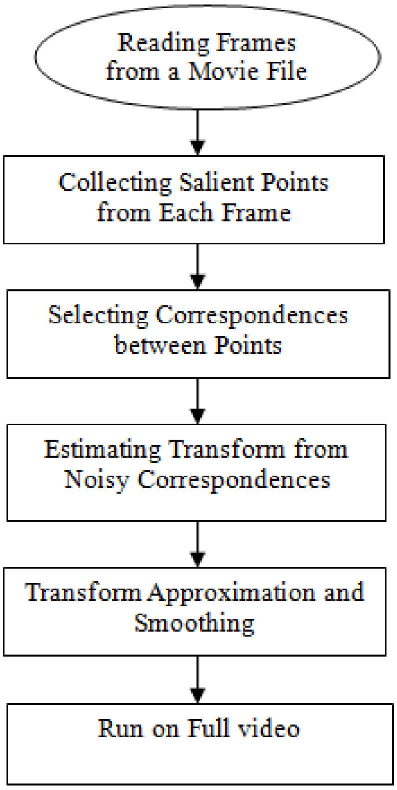


Fig. 2: Flow chart of Proposed Method

A. Feature Extraction and Matching

Extraction of features is one of the very important steps in most of computer vision systems. A good feature extractor is one that extracts only those features that are uniquely recognizable. Here corner based algorithm is used to identify the key points in frames because of variation of intensity values is more at corner comparing to edge and flat surface. Harris Corner Detection algorithm which is one of the fastest algorithms is used to find corner values in frames. The basic idea of Harris Corner Detection is depicted in Fig. 3.

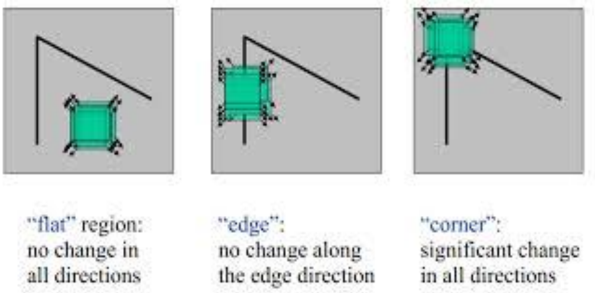


Fig. 3: Basic idea of Harris Corner Detection

After the key points from each frame are obtained, the correspondences between the points that are identified previously need to be picked up. The matching degree between the points that existed in frame A and B is needed to found for each point. The Sum of Squared Differences (SSD) can be used to measure the matching degree between points. Each point in frame A is compared with the corresponding point in frame B to find the lowest matching cost.

B. Motion Estimation

In order to estimate the motion between two consecutive frames an affine transform that makes the descriptors from the first set of points match most closely with the descriptors from the second set is calculated. The affine transform will be a 3-by-3 matrix of the form:

$$\begin{bmatrix} a\_1 & a\_2 & 0 \\ a\_3 & a\_4 & 0 \\ T_x & T_y & 1 \end{bmatrix} \quad (1)$$

Where *a* define scale, rotation and sheering effects of the transform, while T_x and T_y are the translations. However in order to reduce the complexity, a simpler-rotation-translation 2D affine transformation with only four unknown parameters are used to determine the geometrical transformation between two frames. Suppose P(x,y) and P'(x',y') are to be pixel location of corresponding points in consecutive video frames, the relation between these two locations can be expressed as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H.\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

$$H = \begin{vmatrix} S.\cos(\theta) & -S.\sin(\theta) & 0 \\ S.\sin(\theta) & S.\cos(\theta) & 0 \\ T_x & T_y & 1 \end{vmatrix} \quad (3)$$

Where the parameter S represents scaling Theta represents rotation and T_x, T_y represents translation of two successive frames. The parameters of affine transform in equation (3) are estimated based on the method presented in [6] & [7], which utilizes the affine invariant property of the ratio of areas for affine transform parameter estimation.

C. Motion Compensation

To get a stable video, the affine transform obtained in above step is used to compensate the current frame; this is also final step for video stabilization. For a given set of video frames F_i, i=0,1,…,the above procedure is used to calculate the distortion between all frames F_i and F_{i+1} as affine transforms, H_i. Thus the cumulative distortion of a frame *i* relative to the first frame will be the product of all the preceding inter-frame transform as:

$$H_i^{cummulative} = \prod_{j=0}^{i-1} H_i \quad (5)$$

To get a stable video, at each step, the transform *H* between two successive frames is calculated and combined with cumulative transform $H^{cumulative}$. This describes the camera motion henceforth after the first frame. This cumulative transform is used to warp the successive frames in a video.

IV. RESULTS

This section presents the results and discussion of proposed method of video stabilization. The programming was done in MATLAB and later the algorithm was implemented on Raspberry Pi Model B, which is a small

DOI 10.17148/IJARCCE.2015.44101

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

sized computer that plugs into a TV or a Monitor. It contains the ARM processor, which acts as a brain of the device and figures out all the complicated calculations. It also contains Camera Interface Slot (CSI) to interface the 5MP Camera module, which is capable of capturing 1080p video and still images. The Raspberry Pi supports various operating systems and allows Python programming as default language for performing various functions on the device.

A. Key point Detection and Matching

Firstly, the algorithm is developed based on Harris and Stephens corner detection algorithm [4] to detect all key points from each frame. Sample of detected key points obtained from two frames are as depicted in Fig. 4. Next, the correspondence between the key points is identified to reduce the Sum of Squared Differences (SSD) between the consecutive image regions of frames. The descriptors which are used for calculating an affine transformation are depicted in Fig. 5.

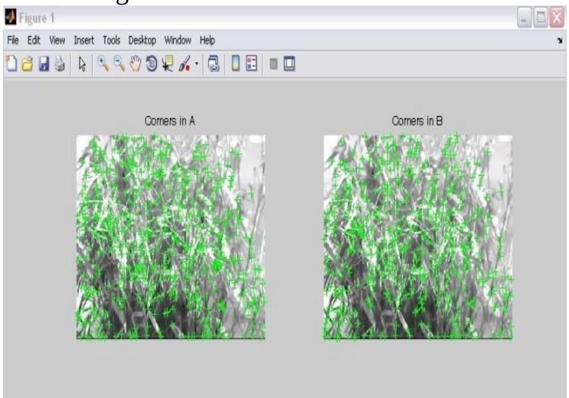


Fig. 4: Key point detection on two consecutive frames

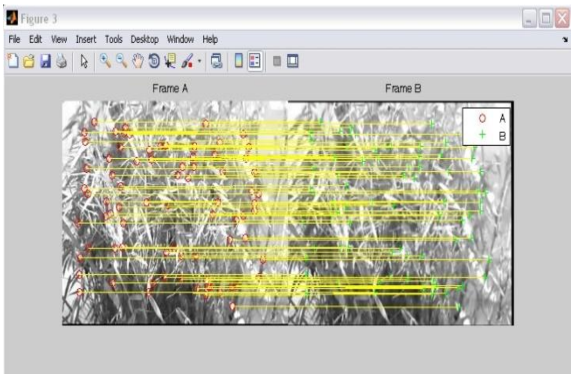


Fig. 5: Key point matching on two consecutive frames

B. Motion Estimation and Compensation

From Fig. 5, it is clear that there are several erroneous key points between two successive frames, but strong estimation of geometric transform between the two frames can be determined using random sample consensus algorithm (RANSAC) [8]. The affine transform estimated from valid corner points is used to project the corresponding points of frame B onto frame A as depicted in Fig. 6. Thus, correcting process stabilizes the video. The proposed algorithm gives good stabilization of frames if the motion between successive frames is less in input video.

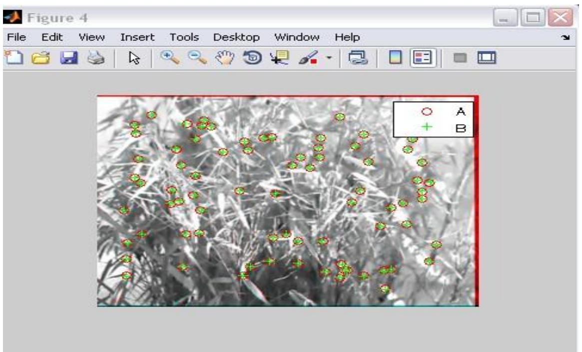


Fig. 6: Correct correspondences according to affine transform

The mean images of original video and stabilized video are depicted in Fig-7. The left image shows the mean of the raw input of 10 frames from original video, while right image shows the mean of corrected frames. It is clear that mean images of original video are more blurred than that of stabilized video.

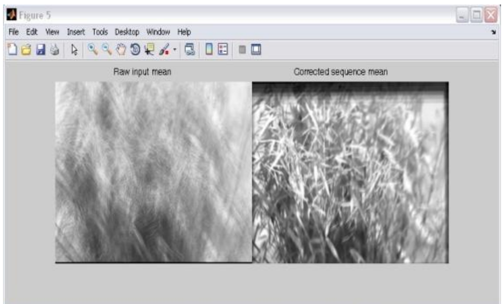
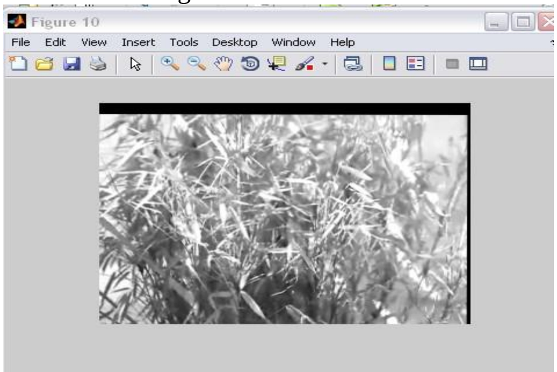


Fig. 7: Mean images of input and stabilized frames

C. Video Output Formatting

From Fig.7, it is clear that the frames of video are stabilized, but some data are missed in the borders which must be filled. This particularly happens when there is a large difference between present frame and first frame. This can be accomplished in several ways. The simplest method is to reset the $H^{cummulative}$ matrix after certain number of frames. However, this method is not efficient for filling the borders in frames. Another method to fill this data is with the old information of the previous video frame(s). However, the combination of both methods will give better stabilization without losing any information from frames as in Fig. 8.



ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

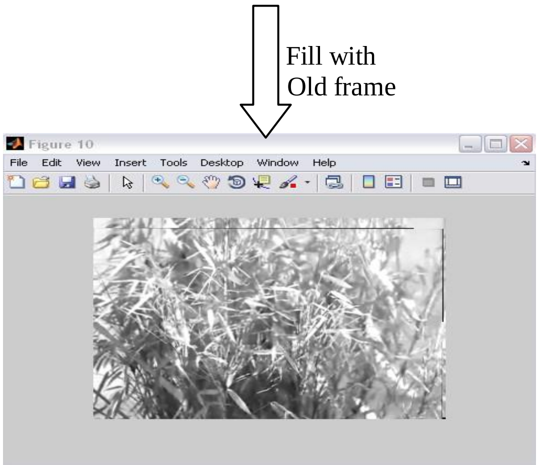


Fig.8: Filling borders with old frame

V. CONCLUSION

In conclusion, the algorithm was initially programmed in MATLAB and later developed on Raspberry Pi using Opencv-Python Programming for real time applications. The video stabilization algorithm that was implemented on Raspberry Pi provides an improvement in the quality of input video by eliminating blurring, jitters, and shakes that exits from the surrounding environment.

ACKNOWLEDGMENT

The authors would like to acknowledge to CSIR- National Aerospace Limited, BMS College of Engineering and TEQIP-II for their support and financial assistance rendered during the tenure of this project work.

REFERENCES

[1] C. Morimoto, R. Chellappa,"Fast electronic digital image stabilization" Proc. 13th International Conference on Pattern Recognition, vol. 3, August 1996.

[2] "Video Stabilization Algorithm Using a Block-Based Parametric Motion Model" report by Ting Chen, Department of Electrical Engineering, Stanford University, USA, 2000.

[3] Anu Suneja and Gaurav Kumar "An Experimental Study of Edge Detection Methods in Digital Methods in Digital Image", Global Journal of Computer Science and Technology, 2010.

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

[5] www.nal.res.in/pdf/MAV.pdf

[6] Y. Wang, Z. Hou and K. Leman, "Combination of Local and Global Features for Near-Duplicate Detection," The 17th Intl Conf on Multimedia Modeling, pp. 430-443, 2006.

[7] Y. Wang, Z. Hou and K. Leman, "Key point-Based Near-Duplicate Images Detection Using Affine Invariant Feature and Color Matching" ICASSP 2011, Prague, Czech Republic, May 2011.

[8] Tordoff B; Murray DW "Guided sampling and consensus for m motion estimation" 7th European Conference on Computer Vision, 2002.

.