A REVIEW ON REAL TIME VIDEO STABILIZATION

Sarang Shirole[1], Prof. A. B. Deshmukh[2]

[1,2]Department of Electronics and Telecommunication, SKNCOE, SPPU, Pune, (India)

ABSTRACT

In the latest technology, there are different applications which are working on videos. Particularly, video from camera's which are placed on moving platform. But due to uneven surface or camera vibration, captured video tends to have undesired jitters, shakes and blurs. This results into unpleasant viewing experience and also affect on video processing application such as surveillance or some military application also. Hence video stabilization technique is essential in such cases. There are different video stabilization techniques are available but they need extra hardware. Hence, Digital video stabilization is an essential video improvement technology which is focusing on removing unnecessary camera vibrations from image sequences by using different digital image algorithms. Again digital stabilization algorithms are of two types, block based and feature based. Features are interesting points in the images and the starting point of many algorithms. Hence it is necessary to have good feature detector and descriptor. In the proposed paper, different feature detector and descriptor pairs are compared based on their speed and number of feature points. Experimental study shows that Features from Accelerated Segment Test (FAST) corner detection algorithm and Fast Retina Key point (FREAK) descriptor are comparatively faster than other pairs. Hence, it is much suitable for real time applications.

Keywords: Feature detection, Feature extraction, video stabilization, MATLAB.

I. INTRODUCTION

Video processing is broadly used in numerous areas such as space, health, auto industry, city planning and military where correctness of image frames are essential. For example, in a medical operation where cameras are used, the doctors need real time stabilized videos to detect the exact position of tumors, cancer cell or any particular disease. Also, military application based on object tracking , successive frames needs to be stable, so that tracking algorithm can work appropriately.

The primary and crucial step of all video stabilization algorithms is to eliminate the noise or undesired effect due to uneven surface. Noise may be in translational and rotational formats. The noises caused in video sequence are associated with the platform on which the video source is located. In the avionics platforms translation and rotational noises can be occurred, however when video source is placed on ground surface where the movement of video source is in spatial directions, the rotational noise is rarely present but translational jitters are very common in such ground application.

In lots of cases, user purposely moves camera to get preferred images. Hence, it is essential to preserve the intentional camera motion while removing the undesired motion due to an unsteady and vibrating platform.

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 10, October 2017
www.ijarse.com
IJARSE
ISSN 2319 - 8354

Combination of camera motion estimation with motion separation determines the undesired motion, which is compensated and thus stable image sequences are produced.

II. RELATED WORK

Existing digital image stabilization algorithms can be broadly classified as block-based [1] [2] [3] or feature-based [4] [5] [6] [7]. Block-based algorithms divide the image into number of blocks, generally squares and these blocks are matched between contiguous frames to obtain local motion vectors, which are used to find a global motion. These algorithms can provide good results, but typically engage heavy computation because of huge numbers of blocks. The solution for this can be feature-based algorithms. The idea behind this is to collect features from each frame and then these features are matched between contiguous frames to calculate motion. The results of Feature-based approaches are more accurate with less computational load, which guarantees a more effective video stabilization solution.

There is no exact definition for a feature, and it is often depends on the problem or the type of application. In general, a feature can be defined as an interesting part of an image [22]. Image features can be edges, corners/interest points, blobs/region of interest points, color and texture etc.

Global features and local features are the two types of image features that can be extracted from image. Global features (e.g., color and texture) intend to describe an image as a whole and can be interpreted as a particular property of the image involving all pixels [21]. While, local features focuses to detect key-points or interest regions in an image and describe them. In this context, if the local feature algorithm detects n key-points in the image, there are n vectors describing each one’s shape, color, orientation, texture and more. The use of global color and texture features are successful for finding related images in a database, while the local structure oriented features are considered more useful for object classification or finding other occurrences of the same object or scene. Meanwhile, the global features cannot distinguish foreground from background of an image, and mix information from both parts together.

On the other hand, the real time applications have to handle large data or to run on mobile devices with limited computational capabilities, there is a growing need for local descriptors that are fast to compute, fast to match, memory efficient, and yet exhibiting good accuracy. Additionally, local feature descriptors are proven to be a good choice for image matching tasks on a mobile platform [21], where occlusions and missing objects can be handled. For certain applications, such as camera calibration, image classification, image retrieval, and object tracking/recognition, it is very important for the feature detectors and descriptors to be robust to changes Image Features Detection, Description and Matching in brightness or viewpoint and to image distortions (e.g., noise, blur, or illumination). While, other specific visual recognition tasks, such as face detection or recognition, requires the use of specific detectors and descriptors.

III. MOTIVATION

Frequently used methods for feature detection and description such as Scale-Invariant Feature Transform (SIFT) [8] and Speeded-Up Robust Features (SURF) [9] are more popular for their good performance. However, these methods have high computational complexity and required comparatively more time. Hence they are difficult to

IJARSE
ISSN 2319 - 8354

implement on embedded systems for real-time applications. FAST feature detector and FREAK descriptor are an efficient local feature detector-descriptor pair that is rotation-invariant and resistant to noise. It was designed using the Features from Accelerated Segment Test (FAST) key-point detector [10] and Fast Retina Key-ponit (FREAK) descriptor [11], leading to an outstanding reduction in computational complexity and hardware cost.The below table shows comparison of well known combination of feature detector-descriptor pairs [15].

Table I

Performance comparison of different combinations of detectors and descriptors

Method	Keypoints	Time (msec)	Time/Point
SIFT+SIFT	3665	5.989	0.0016341
SURF+SURF	3634	1.083	0.0002977
MSER+SIFT	323	0.889	0.0027533
BRISK+FREAK	466	2.531	0.0054322
BRISK+BRISK	466	0.235	0.0005046
ORB+ORB	500	0.236	0.0004715
FAST+BRIEF	11880	0.083	0.0000070

IV. PROBLEM DEFINITION

Implement different feature detector and descriptor pairs on MATLAB and compare their performance.

V. SYSTEM OVERVIEW

The real time images and stored videos are provided to the system. The system executes the different feature detection algorithm and descriptor algorithm. The proposed system consists of five major procedures: video input, frame extraction, feature detection, feature description, video output.

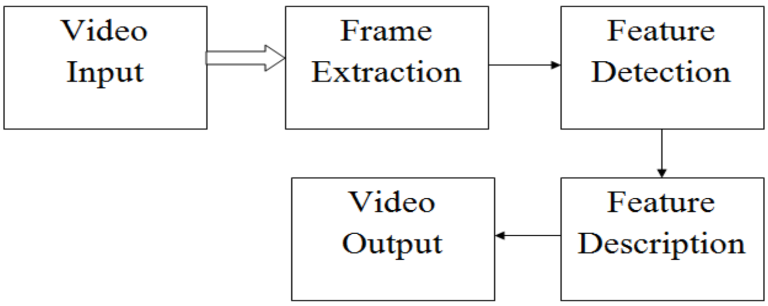


Fig 1: Block diagram of video stabilization system

Video input: The first block is input block which support different kind image format and video format as shown in below table based on operating system installed in your computer. Here images and videos are read as intensity image, since our main algorithm works on intensity values of each pixel.

Table II

Supported Image and video format[12]

Platform	Supported File Name Extensions

IJARSE
ISSN 2319 - 8354

All Platforms	AVI (.avi)
Windows	Image: .jpg,.bmp
	Video: MPEG (.mpeg), MPEG-2 (.mp2), MPEG-1.mpg MPEG-4, including H.264 encoded video (.mp4, .m4v) Motion JPEG 2000 (.mj2) Windows Media Video (.wmv,.asf, .asx, .asx) and any format supported by Microsoft DirectShow 9.0 or higher.

Frame extraction: In this step, frames are extracted from the video as step response.

Feature detection: In this step, the corners of each frame are detected using Features from accelerated segment test (FAST) corner detector. The FAST detector determines whether a pixel is a corner by comparing its intensity value with those of surrounding pixels in a circular ring. To identify whether a candidate point p is actually a corner or not, FAST corner detector uses a circle of 16 pixels (a Bresenham circle of radius 3[13]). Every pixel in the circle is labeled from integer number 1 to 16 clockwise as shown in fig 2. If a set of N adjacent pixels in the circle are all darker than the intensity of candidate pixel p (denoted by I_p) minus a threshold value t or all brighter than the intensity of candidate pixel p plus threshold value t, then p is classified as corner. For a given intensity threshold t, pixels around the centre point can have one of three states [14]:

$$I_i \leq I_p - t \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (Darker)$$

$$I_p - t < I_i < I_p + t \ldots\ldots\ldots. (Similar)$$

$$I_i \geq I_p + t \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (Brighter)$$

where *Ip* and *Ii* denote the intensity value of the center point and of a surrounding pixel, respectively.

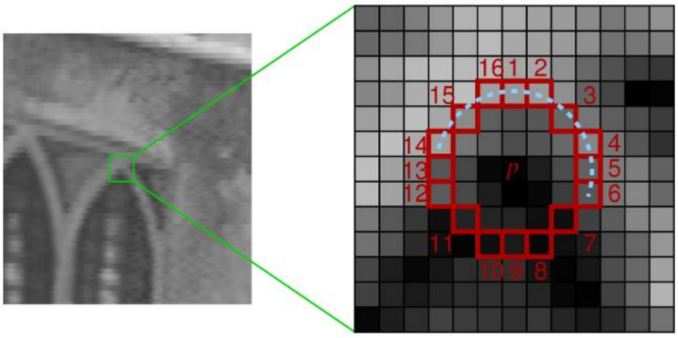


Fig 2 : Pixel allocation for FAST

Feature description: once the corner points are detected, it is necessary to get descriptor for each corner point, for that we have a novel key point descriptor motivated by the human visual system and more specifically the retina, named as Fast Retina Key point (FREAK). By efficiently comparing image intensities over a retinal sampling pattern, a cascade of binary strings is calculated. When compare to SIFT, SURF or BRISK, FREAKs are in-general faster to calculate and required lower memory. Experiments show that FREAKs are more robust

IJARSE
ISSN 2319 - 8354

than other methods. They are thus competitive alternatives to existing key points in particular for Embedded applications [11].

VI. EXPERIMENTAL RESULTS

Proposed algorithm is tested on a laptop with an Intel i5 processor with 2.4 GHz frequency and 4 GB (3.97 GB usable) RAM. Machine runs on 64 bit operating system. The code for proposed algorithm is run on MATLAB R2016a. Proposed algorithm is tested for different types of videos and real time images with different types of scenarios. 2 videos are considered for illustrations, their properties are listed below in table III and their frames are given in fig 3.

We have calculated execution time for feature detection, description and matching for all the test video sequences, as shown in table III. fig 4 shows number of feature points and fig 5 shows execution time required. Proposed detector and descriptor pair gives good execution time with compared to other method. Also we can increase the speed of execution by setting threshold to higher value but it provides less number of detection points which leads to decrease in stabilization.

Table III. Properties of test video

Test video	Resolution	Frame Rate (fps)	Type
a	640x480	59	.avi
b	1280x720	29	.mp4

Video (a) Video (b)

Fig. 3: Frames of test video sequences

Table IV. Comparison of different methods for test sequences Video (a)

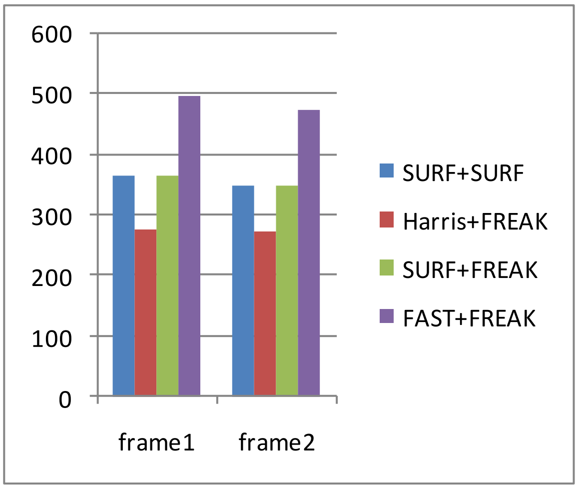
Methods	No. of detected feature points		No. of descriptor feature points		Execution time (sec)
	Frame 1	Frame2	Frame1	Frame2	
SURF+ SURF	365	348	365	348	0.379
Harris+FREAK	278	275	258	259	0.578
SURF + FREAK	365	348	365	348	0.363
FAST+ FREAK At pth = 0.01	1161	1015	1070	945	0.362
FAST+ FREAK At pth = 0.05	978	942	910	884	0.343

IJARSE
ISSN 2319 - 8354

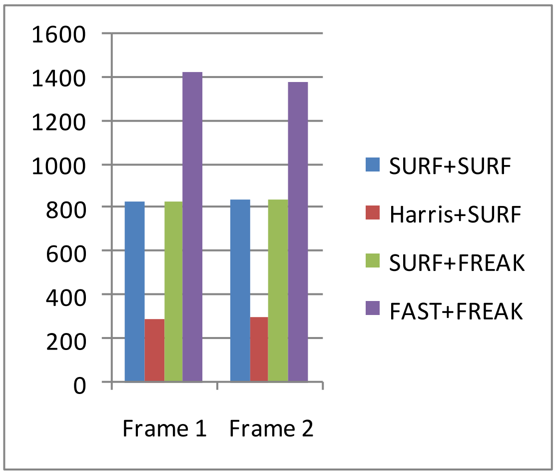
FAST+ FREAK At pth = 0.1	**497**	**476**	**471**	**456**	**0.324**
FAST+ FREAK At pth = 0.2	217	200	210	194	0.308
FAST+ FREAK At pth = 0.3	59	56	59	55	0.295

Video (b)

Methods	No. of detected feature points		No. of descriptor feature points		Execution time (sec)
	Frame 1	Frame 2	Frame 1	Frame 2	
SURF +SURF	830	838	830	838	0.791
Harris + FREAK	288	300	288	300	0.909
SURF + FREAK	830	838	830	838	0.752
FAST+ FREAK At pth = 0.01	1721	1869	1690	1825	0.548
FAST+ FREAK At pth = 0.05	1721	1869	1690	1825	0.548
FAST+ FREAK At pth=0.1	**1427**	**1377**	**1403**	**1352**	**0.532**
FAST+ FREAK At pth = 0.2	456	448	452	446	0.521
FAST+ FREAK At pth = 0.3	212	211	212	211	0.514



Video (a)



Video (b)

Fig 4: Comparison of different methods for test sequences

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 10, October 2017
www.ijarse.com
IJARSE
ISSN 2319 - 8354

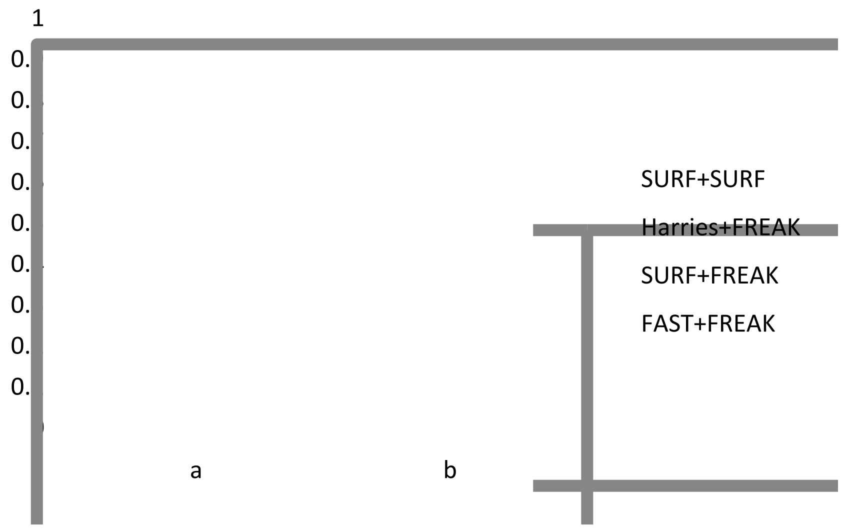


Fig 5 : Execution time for different methods

VII. CONCLUSION

Due to cameras mounted on moving platform, results into undesired jitters making captured video blurred which affects the viewing experience and video processing applications such as video encoding and video surveillance. Hence video stabilization technique is essential in such applications. There are different video stabilization techniques available but they required more hardware and has complex computations. So we conclude that digital video stabilization technique is perfect for real time video stabilization. So in this paper different pairs of feature detector and descriptor have been compared and their implementation on MATLAB with results has been showed. Experimental results show that the combination of FAST corner detector and FREAK feature descriptor is comparatively faster than other. To verify the effectiveness of the proposed system, we conducted the experiments on recorded video data and concluded that compared to SURF+SURF combination proposed FAST algorithm detects 36.16% more corner points and 14.51% faster, similarly 78.77% more corners are detects than Harris + FREAK with 43.94% higher speed. Due to high speed of FAST + FREAK combination it is very effective for real time applications.

VIII. FUTURE WORK

FAST and FREAK descriptor are faster and hence more suitable for real time applications like real time video stabilization.

REFERENCES

[1] S. Battiato, A. R. Bruna, and G. Puglisi, "A robust block-based image/video registration approach for mobile imaging devices," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 622-635, 2010.

[2] G. Puglisi, and S. Battiato, "A robust image alignment algorithm for video stabilization purposes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1390-1400, 2011.

[3] F. Yu, M. Hui, W. Han, P. Wang, L.-q. Dong, and Y.-j. Zhao, "The application of improved block-matching method and block search method for the image motion estimation," *Opt. Commun.*, vol. 283, no. 23, pp. 4619-4625, 2010.

[4] S.-K. Kim, S.-J. Kang, T.-S. Wang, and S.-J. Ko, "Feature point classification based global motion estimation for video stabilization," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 267-272, 2013.

[5] C. Song, H. Zhao, W. Jing, and H. Zhu, "Robust video stabilization based on particle filtering with weighted feature points," *IEEE Trans. Consum. Electron.*, vol. 58, no. 2, pp. 570-577, 2012.

[6] J. Yang, D. Schonfeld, and M. Mohamed, "Robust video stabilization based on particle filter tracking of projected camera motion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 945-954, 2009.

[7] K. Tashiro, and W. D. Reynolds, "Robust and compact infrared video motion stabilization for long-range surveillance," in *Proc. SPIE DSS*, 2013, pp. 87110O-87110O-12.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vision Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.

[10] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105-119, 2010.

[11] Alexandre Alahi, Raphael Ortiz, Pierre Vandergheynst "FREAK: Fast Retina Keypoint" DOI 10.1109/CVPR.2012.6247715 in IEEE CONFERENCE on Computer Vision and Pattern Recognition (CVPR), 2012.

[12] https://in.mathworks.com/help/vision/ref/vision.videofilereader-system-object.html;jsessionid=24fafce827403ac5849706356c04

[13] https://en.wikipedia.org/wiki/Features_from_accelerated_segment_test

[14] Jianan Li, Tingfa Xu, and Kun Zhang, "Real-Time Feature-Based Video Stabilization on FPGA". DOI 10.1109/TCSVT.2016.2515238, IEEE Transactions on Circuits and Systems for Video Technology, 2015.

[15] Sahin Isik and Kemal Ozkan, " A Comparative Evaluation of Well-known Feature Detectors and Descriptors" International Journal of Applied Mathematics, Electronics and Computers ISSN: 2147-82282 Accepted on 15th August 2014.

[16] P. Hsuao, C. Lu, and L. Fu, "Multilayered image processing for multiscale Harris corner detection in digital," IEEE Trans. Ind. Electron., vol. 57, no. 5, pp. 1799–1805, 2010.

[17] C.-K. Liang, Y.-C. Peng, H.-A. Chang, C.-C. Su, and H. Chen, "The effect of digital image stabilization on coding performance [video coding]," in *Proc. ISIMP*, 2004, pp. 402-405.

[18] C. Morimoto, and R. Chellappa, “Evaluation of image stabilization algorithms,” in *Proc. ICASSP*, 1998, pp. 2789-2792.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in *Proc. ICCV*, 2011, pp. 2564-2571.

[20] https://en.wikipedia.org/wiki/Image_stabilization

[21] M. Hassaballah, Aly Amin Abdelmgeid and Hammam A. Alshazly “Image Features Detection, Description and Matching” Springer International Publishing Switzerland 2016, DOI 10.1007/978-3-319-28854-3_2.

[22] https://en.wikipedia.org/wiki/Feature_detection_(computer_vision)