Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan Vol. 22 No. 4-2017
ISSN:1813-1786 (Print) 2313-7770 (Online)

Video Target Stabilization Based on Motion Vector Compensation

B. Tariq[1], M. Amjad[2], S. A. Niazi[3], M. I. Malik[4]

[1,2,3]*Electronic Engineering Department, The Islamia University of Bahawalpur*
[4]*Computer Engineering Department, BZU, Multan.*
[1]bismah.tariq91@gmail.com

Abstract-In this paper a solution is proposed for stabilizing the target in un-stabilized shaky videos by using the motion vector compensation technique mainly focusing on template-based approach for locating target in desired frames. Frames are extracted from the input video and are formatted for further processing. Noise filtration is then done to neutralize the affects of inherent noise in the input video. Dimensions and other parameters are extracted from the input frames and the target is then selected. The target is matched in successive frames using cross correlation matching. Center positions are then updated which are later used for motion vector calculation and offset updating. Finally, the effects of un-necessary target and camera movements are removed by linear translations. Modified frames are then sequentially stacked for video reconstruction and display. The proposed algorithm is useful in security applications and in research works like astrological studies and the study of heavenly bodies.

Keywords-Frames Per Second, Motion Vector, Computer Vision Toolbox, Picture Element, Noise To Signal Ratio, Cross Correlation.

I. INTRODUCTION

In todays modern age, security cameras are being widely used in almost every organisation, institute, as well as on moving air and ground vehicles, for the porpose of enhanced security. There arise the problem, with the captured video that the output is of low quality due to slow and undesired motions like zoom, pan and tilt etc. In such scenarios, stabilization of captured results is required which is achieved by estimating and removing the inter frame motion, causing degradation, between the successive frames. Video stabilization is the technique used to remove the unwanted camera motions and hand shaky movements from video.

Video stabilization aims to reduce the high frequency frame to frame jitter that is produced by the camera motion [i]. This problem arises due to presence of camera placed on moving platform. Stabilization is important in fields such as enhancement and restoration, improve contrast and remove blurring, remove scratches from an old movie or image, security and right protection, object or target focusing, information analysis and automatic recognition.

A mechanical stabilization system includes a vibration from the sensors (gyros accelerometer) [ii]. That has been used in early stages of camcorder. Optical image stabilization uses moveable or prism lens assembly that adjusts the path length of light when it travels in camera lens system [iii]. Video stabilization algorithm follows the sequence of interframe motion esimation, motion smoothing or filtering and finally motion compensation as shown in Fig 1.

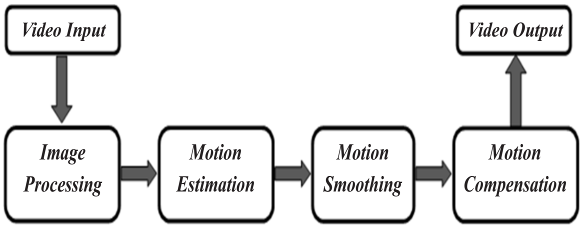

Fig. 1. Video Stabilization Scheme

Motion estimation relies on motion vectors calculation that may be related to part of image or whole image. Pel-recursive approach is one of the most early technique used to calculate approximate motion to establish video coding. Basically, in these methods, displacement is estimated recursively to minimize the Displaced Frame Difference (DFD). Though, the pel-recursive techniques are comparatively easier to put into practice, however their convergence might be time taking, also the motion estimation may not always be of very fine quality. It is particularly observed in the cases where displacement is a vital parameter, or in the cases where there are huge discontinuities in motion that too at the borders of the object. This restraint is typically because of the nature of this algorithm that is recursive, due to the causality constraint.

The output of block matching algorithm contains the motion vectors for each block and have pixels value difference between blocks in current frame and matched block in reference frame [iv]. The main idea behind the matching is to divide current frame into matrix of macro blocks of frame [v]. These blocks are compared with corresponding block and its adjacent neighbors will create a vector that shows movement of macro block from one location to another in previous frame. This movement estimates motion in current and previous frame.

While on the other side, phase correlation method determines the movement between the two fields from their phases and hence estimates relative offset between consecutive frames. Phase correlation method is based on fourier transform and is insensitive to variation of illumination.

Optical flow technique calculates the motion of pixels of an image sequence and gives a point to point correspondence. It provides a vector field that represent the 3D motion of object points across the 2D image, where displacement vector show certain pixels position that points to where that pixels can be established in reference frame. Target template is manually cropped from the first frame. The same template is then matched automatically in successive frames by processing of template matching algorithm. The desired target template is selected in the initiale phase. This template is the target area in the captured video around which the stabilization is required. In the projected proposal, motion vectors are calculated from the extracted parameters [vi].

Motion smoothing helps to smooth out the unwanted camera motions by applying a low pass filter on motion vectors for removing the high frequency components that may cause bluring of sharp edges of image and thus its application is limited to a certain extent. Motion compensation is applied to perform transformation of a reference frame to current frame. As per proposed model, motion compensation is accomplished by inserting a counter effect of the offset on the frames that neutralise the shaky effects of the target.

II. VIDEO STABILIZATION

The whole scheme comprises of three phases; *(i)* initialization phase, *(ii)* processing loop and *(iii)* result display. The working scheme of video target stabilization is as per Fig 2. In the *initialization phase,* the video is fed as input. Video source is either an already stored file in the drive, or it can be a live stream from an integrated camera/capturing device. Video is decomposed into frames at standard frame rate, i.e., 30 frames per second (fps). Standard frame rate of 30 fps has been used throughout in this work. This has been already mentioned in the paper. NTSC (National Television System Committee) video in North America and Japan uses 29.97 fps. Other common frame rates are usually multiples of these. Some high definition cameras can record at 30 fps, as opposed to 29.97 fps. However, before color was added to NTSC video signals, the frame rate was truly 30 fps. Format of all video frames is converted from RGB to Grayscale. De-noising and enhancement filters are implemented where required. Although De noising and enhancement is optional, but without it the resulting video quality will be poor. This info is also added in the paper. Target template is selected by cropping and its basic information/parameters are extracted such as, template's center position, origin, size and dimensions. There is no constraint on the size of cropped area for target selection. *Processing loop,* runs for the number of iterations equal to the number of frames. On every iteration, pre-selected target template is matched and calculations are made for parameters like motion vector, offset, etc. The image linear translation and zero padding is done on the processed frames. Image linear translation is achieved by pixel coordinates shifting. It is implemented by counter balancing the effect of image offset pixel by pixel. Pixels are shifted in the opposite direction as that of the image offset value obtained. Stabilized frames are obtained after the image translation and zero padding. Zero-padding will have a small effect on the quality of final video. However zero-padded areas will appear in black color on almost all four boundaries of the video. After the linear translation of the frame, the stabilized frames are obtained of different sizes. However we want to reconstruct the output video in the end from the stabilized frames. For video reconstruction purpose, all the stabilized frames must be of equal size. Thus zero padding has been done to obtain same sized frames. For the *result display,* all stabilized equal sized frames are run in the display window at standard frame rate.

III. TEMPLATE BASED STABILIZATION

Video target stabilization technique is based on motion vector compensation that makes use of template-based approach for template matching. The selected template corresponds to the desired target in the video that is needed to be stabilized and focused and hence it is the first mile stone to extract useful information like target's origin, size, center, center position and the template search border. Template matching is done to find small parts of an image that match the given or the required template image.

If the template consists of strong features like contours, shape descriptors, textures, invariant moments etc, then a feature based approach is preferable for template matching. Features of an image are specific structures in the image such as points, edges or objects. By strong features of the template we mean the sharp edges which are result of high frequency components. It is computationally efficient when the source images are of larger resolution, as it doesn't require the entirety of the template image. While template-based approach is efficiently useful for the templates lacking strong features, or when the bulk of template image consists of the matching image. Since template-based technique needs sampling of a large number of points to search in order to find the best match location, it is possible to reduce the sampling points by reducing resolution of search and the template images by the same factor and then performing the operation on the downsized images [vii]. This method provides a search window of data points in the search image and has the advantage that the template doesn't need to search all viable data points. The summarization of implementation steps of template-based matching approach is shown as a block diagram in Fig 3.

When the two functions ***f*** and ***t*** are complex valued functions and one of those is being conjugated, it ensures that the aligned peaks or the aligned troughs of the functions will have the imaginary components. Hence these will positively contribute in the integral, while their best match is being sited. The XCORR is an easy and efficient way to find the matching location of the template image in the frame or source image [viii]. XCORR is used for rows and columns of matrices to get 2D results. In the language of signal processing, the cross correlation is actually the measure of similarity of the two waveforms, as a function of time-lag applied to one of the waveform. The other names of cross correlation are sliding dot product or the sliding inner-product. Its purpose is to search a long signal (a frame) for a short, known feature (a template).

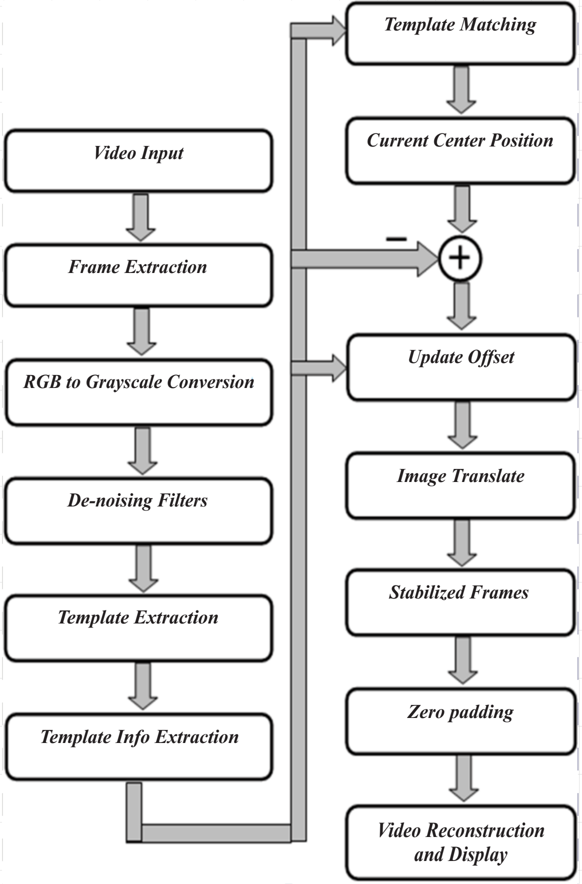

Fig 2: Proposed Stabilization Model

In case of the continuous signals or functions f and g, we can define the cross-correlation by the Eq. 1.

$$(f*g)(t) \overset{def}{=} \int_{-\infty}^{\infty} f^{*}(\tau)\, g(t+\tau)\, d\tau \qquad (1)$$

Where in this equation, the satiric sign (*) with the function f indicates the complex conjugate of f.

Similarly in the case of two discrete functions, the mathematical equation of cross-correlation can be written in equation (2)

$$(f*g)[n] \overset{def}{=} \sum_{-\infty}^{\infty} f^{*}[m]\, g[n+m] \qquad (2)$$

The nature of cross-correlation is same as that of the convolution of two functions except that convolution requires flipping of one of the functions. In this technique, a convolution mask is used, that contains the specific feature of the anchor frame or the search image, and the result of convolution is maximum at the point where the image structure and the mask structure have the best match, i.e. at the location where the large image values are multiplied with the large mask values [ix]. After each iteration, the maxima and the corresponding subscripts are recorded for motion vector and hence the required offset calculation and updation.

After the template is matched and subscripts are recorded, the templates new origin, center and center position on the anchor frame can be computed by some simple mathematical calculations. As it is previously discussed that the maximum value is at the point where the template's center best matches the frame pixels below. So from that center coordinates, we move to the origin of the template by subtracting half of the template height and width from the rows and columns respectively.

Next parameter is the template center which is simply the half of the template's total size. The template's center position is calculated by making the use of template's origin and center calculated in previous steps. The center position is the sum of the template's origin and the template's cente. While both of these are already calculated. Template tracking is mainly based upon calculation of correlation matrix. Highest correlation index indicates that the template is matched. Further motion vector is then calculated for offset calculations. The template is manually selected only once from the initial frame. All the information is then extracted by algorithm processing. The template and its center position is matched in the successive frames by the processing of template matching algorithm. Template matching algorithm is implemented using cross correlation and same has been illustrated in Fig. 3. Thus no human intervention is involved for tracking the target template center position in successive frames. Zero padding is required to make ensure that all the stabilized frames are of same size. This is essential for video reconstruction from stabilized frames.

IV. MOTION VECTOR AND OFFSET CALCULATION

The most important element in any process of the motion estimation is the motion vector. By the numerical value of motion vector, the observer can easily tell that whether the motion is performed or not and can also estimate it. Its purpose is to represent a block or a macro block (a small block portion in the whole frame), in an image, depending upon the position of the similar block or the macro block within another frame in the sequence (mostly the previous one), which is often named as reference frame or the reference picture [x].

More precisely and technically, the motion vector of an image can be defined as a 2D vector, which is intended for the inter prediction that gives the offset value from the coordinates in the existing frame of the sequence, to that of the coordinates in reference frame, i.e. the frame previous to the current frame in the sequence.

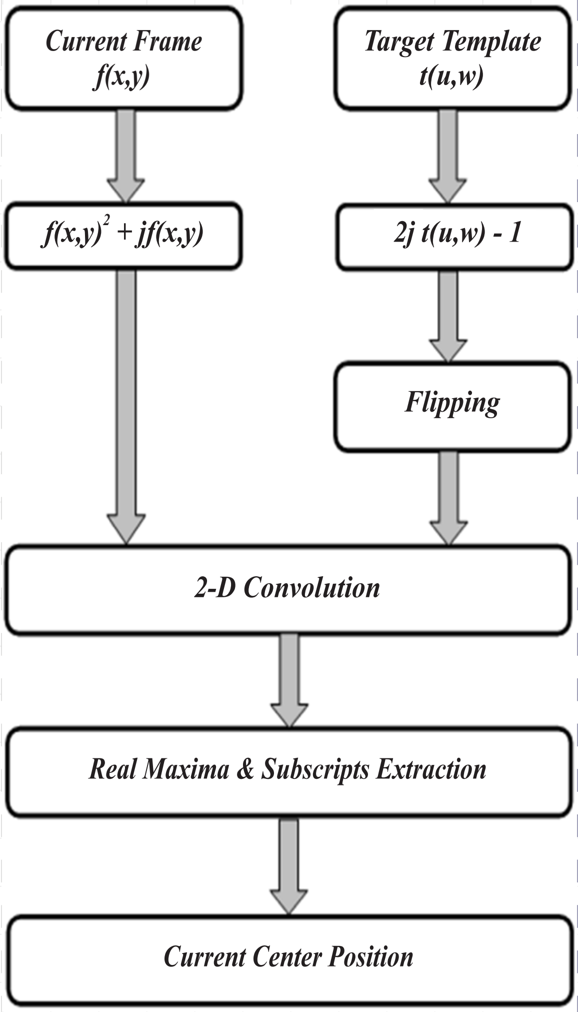

Fig. 3. Template Matching Algorithm

Now by keeping the above definition and discussion in mind, the motion vector can be mathematically explained as the difference of the center pixel location of that block or the macro block which we are considering in the existing frame, and center pixel location of same block or macro block within the preceding frame. However, in our project code, that block is actually the target template, which we have selected manually. This scheme may be described in the form of a block diagram in Fig. 4.

In general, by offset we mean the displacement between two specified points or destinations. In the image processing context, the meaning of the image offset is the displacement of the coordinates of the target or some template from one frame to the next in a video sequence. Such as in present case, we want to find our target's inter-frame offset. Depending upon this target's coordinate offset, the image translation can be performed to stabilize our target in the video. The offset is first initialize to *[0; 0]* for the very first frame and after that it is constantly updated for each frame transition, by the execution of an offset update algorithm [xi].

The implementation of the offset update algorithm is accomplished by some mathematical steps and equations constructed by making the use of some template information, motion vector and the previous offset value. The offset update algorithm can be elaborated by the block diagram shown in Fig. 5.

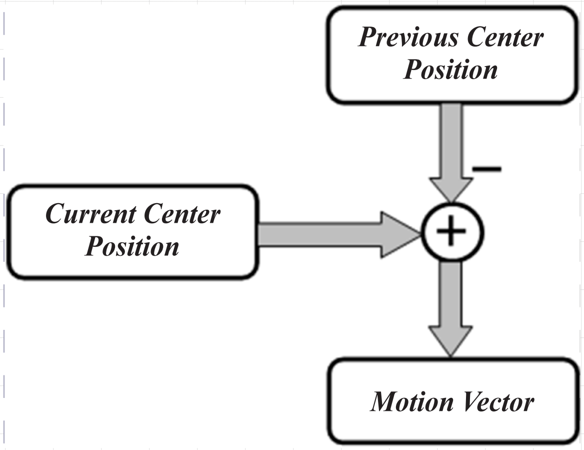

Fig. 4. Motion Vector Calculation

Depending upon the calculated motion vector and offset coordinates, necessary linear translation is performed on each frame and zero padding of all frames is done upto desired level for the purpose of video reconstruction from the stabilized frames. It is required because the linear translation in either direction creates a blank area in the opposite direction. This padding is applicable to all video formats and hence is useful in any applications. Zero padding is the appending of zeros to a signal or an image. It modifies a signal of length N to an extended signal of length M>N. This can be stated in the form of Eq. 3:

$$ZeroPad_{M,M}(x) \overset{\Delta}{=} \begin{cases} x(m), & 0 \le m \le N-1 \\ 0, & N \le m \le M-1 \end{cases} \quad (3)$$

The purpose of zero padding is accomplished by constructing a null matrix of the video size and then over riding the stabilized frame onto it [xii].

V. COMPUTATIONAL RESULTS

Computation of motion vector and the required inter frame offset and translation is done by the basic but most important information extracted out of the template or the target selected. Table 1 illustrates the information extracted and computed from the target template image during the first fifteen processing loop iterations.

The real maxima and its corresponding indices and subscripts computed as a result of template matching algorithm during first fifteen loop iterations are stated in the Table II.

Motion vector calculation is based on the difference of template center position in two consecutive frames which forms the basis of offset computation. Table III states the motion vectors for the first fifteen loop iterations.

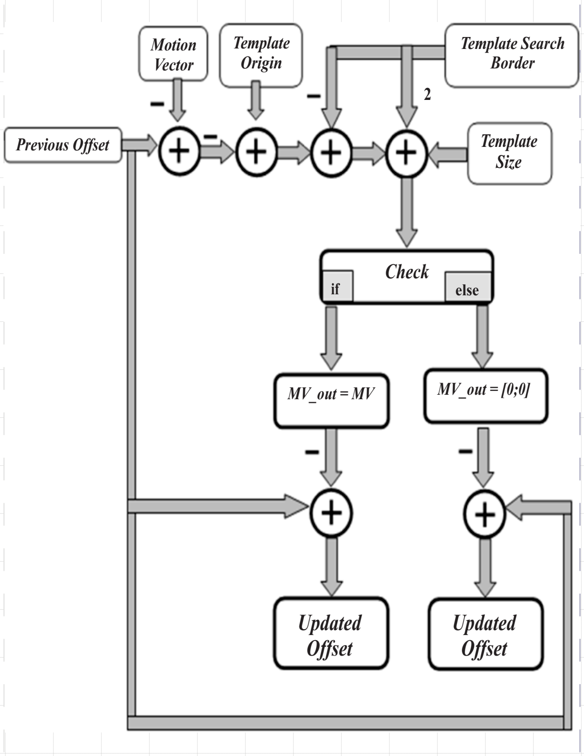

Fig. 5. Offset Update Algorithm

TABLE 1
EXTRACTED TEMPLATE INFORMATION

Iterations	Origin	Center	Center Position
1.	[61 116]	[41 20]	[101 135]
2.	[60 115]	[42 21]	[101 135]
3.	[60 115]	[42 21]	[101 135]
4.	[53 126]	[42 21]	[94 146]
5.	[54 124]	[42 21]	[95 144]
6.	[49 123]	[42 21]	[90 143]
7.	[56 123]	[42 21]	[97 143]
8.	[59 121]	[42 21]	[100 141]
9.	[58 119]	[42 21]	[99 139]
10.	[62 122]	[42 21]	[103 142]
11.	[53 128]	[42 21]	[94 148]
12.	[54 128]	[42 21]	[95 148]
13.	[51 130]	[42 21]	[92 150]
14.	[51 139]	[42 21]	[92 159]
15.	[63 135]	[42 21]	[104 155]

TABLE II
TEMPLATE MATCHING ALGORITHM COMPUTATIONS

Iterations	Real Maxima	Indices	Column	Row
1.	54757759	32502	102	136
2.	54757919	32502	102	136
3.	54735453	35135	95	147
4.	54724794	34656	96	145
5.	54730456	34411	91	144
6.	54722462	34418	98	144
7.	54710758	33941	101	142
8.	54705383	33460	100	140
9.	54687955	34184	104	143
10.	54664570	35615	95	149
11.	54664570	35615	95	149
12.	545464722	35616	96	149
13.	54629655	36093	93	151
14.	54622804	38253	93	160
15.	54613216	37205	105	156

TABLE III
MOTION VECTOR CALCULATION

Iterations	Previous Center Position	Current Center Position	Motion Vector
1.	---	[101 135]	[0 0]
2.	[101 135]	[101 135]	[0 0]
3.	[101 135]	[101 135]	[0 0]
4.	[101 135]	[94 146]	[-7 11]
5.	[94 146]	[95 144]	[1 -2]

6.	[95 144]	[90 143]	[-1 -5]
7.	[90 143]	[97 143]	[7 0]
8.	[97 143]	[100 141]	[3 -2]
9.	[100 141]	[99 139]	[-1 -2]
10.	[99 139]	[103 142]	[4 3]
11.	[103 142]	[94 148]	[-9 6]
12.	[94 148]	[95 148]	[1 0]
13.	[95 148]	[92 150]	[-3 2]
14.	[92 150]	[92 159]	[0 9]
15.	[92 159]	[104 155]	[12 -4]

TABLE IV
OFFSET CALCULATION AND UPDATION

Iterations	***Previous Offset***	***Motion Vector***	***Template Origin***	***Updated Offset***
1.	—	[0 0]	[61 116]	[0 0]
2.	[0 0]	[0 0]	[60 115]	[0 0]
3.	[0 0]	[0 0]	[60 115]	[0 0]
4.	[0 0]	[-7 11]	[53 126]	[7 -11]
5.	[7 -11]	[1 -2]	[54 124]	[6 -9]
6.	[6 -9]	[-1 -5]	[49 123]	[11 -8]
7.	[11 -8]	[7 0]	[56 123]	[4 -8]
8.	[4 -8]	[3 -2]	[59 121]	[1 -6]
9.	[1 -6]	[-1 -2]	[58 119]	[2 -4]
10.	[2 -4]	[4 3]	[62 122]	[-2 -7]
11.	[-2 -7]	[-9 6]	[53 128]	[7 -13]
12.	[7 -13]	[1 0]	[54 128]	[6 -13]
13.	[6 -13]	[-3 2]	[51 130]	[9 -15]
14.	[9 -15]	[0 9]	[51 139]	[9 -24]
15.	[9 -24]	[12 -4]	[63 135]	[-3 -20]

Depending upon this target's coordinate offset, the image translation is decided to stabilize desired target in the video. The offset for the very first frame is initialized to *[0; 0]* and is constantly updated for each frame transition afterwards, by the execution of offset update algorithm. The offset calculation and updated results are enlisted in Table IV.

Relying upon the updated offset, each frame is linearly translated by required coordinates to achieve the stabilized frames which in turn reconstruct the stabilized video. This concludes that the offsets are the undesired motions required to be eradicated.

To compare the results, variance and standard deviation of offset arrays of both input and output videos are computed. Standard deviation and variance of input video are as follows;

$$\sigma_1 = 10.5857 \quad 7.5506 \tag{4}$$

$$\sigma_1^2 = 112.0565 \quad 57.0122 \tag{5}$$

While the results of standard deviation and variance of stabilized output video are as follows;

$$\sigma_2 = 0.1443 \quad 0.1015 \tag{6}$$

$$\sigma_2^2 = 0.0208 \quad 0.0103 \tag{7}$$

Above mentioned comparison results clearly show that the standard deviation and variance of the stabilized video are less than the input video. From these values it is concluded that the output video is 98% stabilized compared to the input destabilized video.

VI. CONCLUSION

PRAs are the repetitive refining of the motion estimation, for the individual pixels, by the gradient techniques. The BMAs presume that all of the pixels inside a certain block, has the similar activity of motion. The BMAs estimate the motion, depending upon the rectangular blocks, and thus they create merely one motion vector, per individual block. The PRAs engross more complications of computations and a lesser amount of regularity, therefore these are tricky to be realized in the hardware. Generally, the BMAs are comparatively more appropriate for realizing an uncomplicated hardware, due to their simplicity and regularity.

The proposed technique for video target stabilization is a simple easy-to-implement technique. It has wide spread uses in many security, research and engineering fields. Moreover, it has the flexibility to be extended in implementation at high security and sensitive areas, for giving better and up to the mark results. Right now this technique is best efficient for video streams having the focused target within the boundaries of the frame and where the target must not exceed the moderate size ranges, otherwise manually selected target is stabilized for better perception of important information in the video data such as vehicle number plate reading, suspicious luggage checking on airport and bus stops, stabilization of astrological videos and in endoscopic videos for medical diagnosis etc.

REFERENCES

[i] L. Zhang, and Q. Huang, "A Global Approach to Fast Video Stabilization," IEEE Transactions on Circuits and Systems for Video Technology, Volume: 27, Issue No 2, Feburary 2017.

[ii] P. Rawat and J. Singhai, "Review of Motion Estimation and Video Stabilization techniques for hand held mobile video," Signal & Image Processing An International Journal (SIPIJ),

Volume: 2, Issue No. 2, June 2011.

[iii] F. L. Zhang, J. Wang, H. Zhao, R. R. Martin, and S. Hu, "Simultaneous Camera Path Optimization and Distraction Removal for Improving Amateur Video," IEEE Transactions on Image Processing, Vol 24, Issue No. 12, December 2015.

[iv] K. Mayen, C. Espinoza, H. Romero, S. Salazar, M. Lizarraga, and R. Lozano, "Real-time video stabilization algorithm based on efficient block matching for UAVs," Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), IEEE, 2015.

[v] L. C. Manikandan and R. K. Selvakumar, "A study on Block Matching Algorithms for Motion Estimation in video coding," International journal of Scientific and Engineering Research, Vol. 5, Issue 7, 2014.

[vi] J. T. Philip, B. Samuvel, K. Pradeesh, and N. K. Nimmi, "A comparative study of block matching and optical flow motion estimation algorithms," Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD), IEEE, 2014.

[vii] S. K. Sahani, G. Adhikari and B. K. Das, "A Fast Template Matching Algorithm for Aerial Object Tracking," International Conference on Image Information Processing, 2011.

[viii] J. N. Sarvaiya, S. Patnaik, and S. Bombaywala, "Image Registration by Template Matching using normalized cross-correlation," International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009.

[ix] F. Essannouni, R. Oulad and D. Aboutajdine, "Simple Noncircular Correlation Method for Exhaustive Sum Square Difference Matching," Optical Engineering, Volume 46, Issue 10, Image Processing, 2007.

[x] P. Liu, H. Wang, and W. Xiaojuan, "Motion Compensation Based Detecting and Tracking Targets in Dynamic Scene," 2010 International Conference on Measuring Technology and Mechatronics Automation.

[xi] A. Hameed, "Video Shot Detection by Motion Estimation and Compensation," International Conference on Emerging Technologies, 2009.

[xii] K. Briechle, U. D. Hanebeck, "Template Matching using Fast Normalized Cross-Correlation," SPIE Proceedings, Volume 4387, Optical Pattern Recognition XII, 95, March 20, 2001.