Embedded Implementations of Real Time Video Stabilization Mechanisms: A Comprehensive Review

Mohammed Ahmed[1], Dr. Laxmi Singh[2]

[1]Research Scholar, [2]Professor,
[1,2]Rabindranath Tagore University, Bhopal, Madhya Pradesh, India

ABSTRACT

Video Stabilization has been widely researched and still is under active research considering the advancements that are being done in the field of digital imagery. Despite all the works, Hardware based Real time Video Stabilization systems; especially works dealing with implementation of Technology on prototype Implementation boards have been very few. This review works focuses on the specific aspect of the modern day Stabilization systems implemented in prototyping boards and the algorithms that have been found suitable for such implementation taking considerations of cost, size and speed as the principal criterions.

KEYWORDS: *Motion Estimation, Real-time Stabilization, FPGA, SIFT, SURF*

How to cite this paper: Mohammed Ahmed | Dr. Laxmi Singh "Embedded Implementations of Real Time Video Stabilization Mechanisms: A Comprehensive Review" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-3, April 2021, pp.685-693, URL: www.ijtsrd.com/papers/ijtsrd39963.pdf

IJTSRD39963

1. INTRODUCTION

Digital Video stabilization has been the more favored technique in recent times as compared to its more accurate but expensive counterparts like mechanical and optical stabilization. This is primarily because of the fact that Software based implementations provide a viable alternative to solutions having costly and bulky hardware assemblies like sensors, gyroscopes, lenses, making DVS based setups portable and cost effective. A generic DVS system consists of three major blocks as shown in fig 1.The first block in Digital stabilization is of Motion Estimation where global motions in the frames of the video are extracted. Then, intentional motions are separated from the global motions in the second step which leads to motion correction stage. The last block deals with image correction to produces the final stabilized video using the estimated unintentional motions.

Motion estimation is the most time consuming and difficult part in digital stabilization which also forms the basis of the operational speeds of the stabilization system. For real time stabilization mechanisms, computational speeds are of critical importance which actually are a function of the employed algorithm being computationally less intensive. Motion estimation, being the most complicated section, needs to be focused on the most for achieving the objectives of real time stabilization. Various mechanisms have been used to implement the motion estimation stage having differing computation speeds and complexity levels, and also with varying degrees of accuracy. The paper discusses the following main things; the first part discusses the working details of Video Stabilization Techniques, the second part discusses the works and stabilization techniques that have been utilized for realtime applications, especially prototyped implementations. The last part discusses the realtime performance of feature based descriptors like SIFT and SURF and the associated advantages with them.

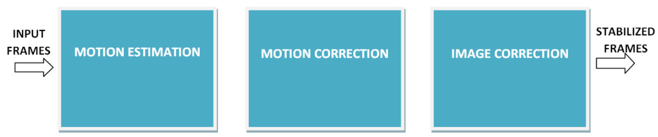

Fig 1 Generic Video Stabilization System

2. Motion Estimation Techniques

The motion Estimation is of five types in general:

1. Gradient based methods
2. Pel Recursive Techniques
3. Block based Matching methods
4. Feature Matching Methods

2.1. Gradient techniques

For image sequence analysis based applications, gradient techniques can be employed. Gradient techniques make use of the optical flow constrain equation. [1,2]

$$\vec{v}.\vec{\nabla} I(\vec{r},t) + \frac{\partial I(r,t)}{\partial t} = 0 \quad (1)$$

in order to find the motion vector ~v on the position ~r with some additional constraints. For example, a Horn-Schunck method [3] minimizes the square of the optical flow

$$\left(\frac{\partial v_x}{\partial_x}\right)^2 + \left(\frac{\partial v_x}{\partial_y}\right)^2 \text{ and } \left(\frac{\partial v_y}{\partial_x}\right)^2 + \left(\frac{\partial v_y}{\partial_y}\right)^2 \quad (2)$$

Some preliminaru works employing gradient mechanisms have been done by Toshiaki Kondo, Pramuk Boonsieng et.al [4,5] wherein thet discussed two conventional gradient based motion estimation techniques. Since Unit gradient vectors have the advantage of being insensitive to constantly varying image intensity, the methods suggested by [4,5] were further improved in [5] by using unit gradient vectors rather than utilizing image intensities. The works resulted in better motion estimation techniques more robust to irregular lighting conditions as compared to conventional systems.

In [6], information from consecutive frames was uses to account for the trans-rotational motions. The work used an optical flow based affine model for motion estimation using the Horn Schunck algorithm. A model fitter was the used to stabilize video sequences afterwards.

2.2. Pixel recursive Techniques

As a variation of gradient techniques, Pixel recursive techniques were introduced. It is basically an iterative gradient mechanism which minmizes predictive error using Displaced Frame Difference (DFD). It owes its roots to the Netravali Robbins method [7], which works on the principle of recursively updating the DFD vector according to the formula

$$\vec{d}^{k+1} = \vec{d}^{k} - \in \mathrm{DFD}(\vec{r},t,\vec{d}^{k}).\nabla_{\vec{r}} I(\vec{r}-\vec{d},t-\Delta t) \quad (3)$$

2.3. Block Matching Techniques

Block Matching Techniques [8] work on the minimization of a differentiative measure. In other words, blocks in current frame are matched with with those in previous frame. The best prediction is done by matching between the current block and all blocks in the search area, also known as full search algorithm. If the algorithm uses MSE as parameter for matching, then for every block size of 16×16, the algorithm will require 256 subtractions, 256 multiplications and 255 additions which is fairly resource exhaustive [9]. This is the primary reason the algorithm is rarely used, especially in real time applications.

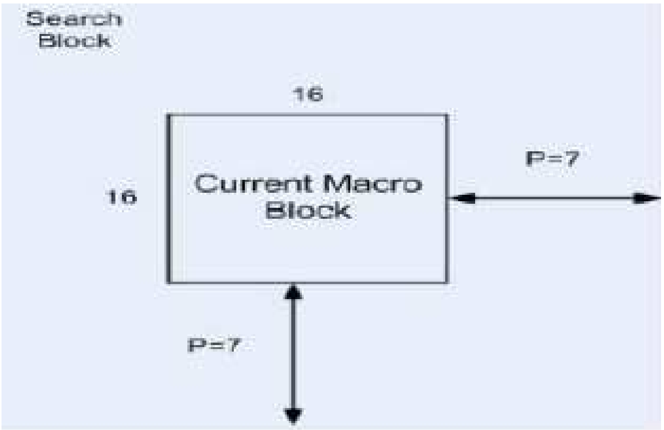

Fig 2 Block matching a macro block of size 16 × 16 pixels and a search Parameter p of size 7 pixels.

Three step search (TSS) is a moderated version of the full search algorithm, with a search location at the center of the search area and searches in search window with sides of 4. Three motion estimation algorithm have been used in [10, 11]. The authors in [10] suggested a proposed a modified adaptive rood path search (N-ARPS) algorithm with small motion prejudgment (SMP), where it will decide on the block search strategy taking into account motion properties. Computational efficiency in [12] was improved using the three step search (TSS) along with GCBP matching that performed a competent search during correlation measure calculation. Another variation of block based searching uses Hexagon based search algorithm (HEXBS) [14] which is based on the same search pattern as in diamond search but instead uses a hexagon in place of diamond search pattern. An adaptation of the HEXBS was used by [14] wherein the direction of motion was predicted by a rood shaped pattern incorporated with Hexagonal based Search (HexBS) for refining search process.

In context of realtime implementations, BBGDS (block based gradient descent search) and diamond search (DS) have been seen to outperform other algorithms in terms of lesser number of computations. The DS algorithm consists of a small diamond search pattern (SDSP) and large diamond search pattern (LDSP). The LDSP pattern is targeted at finding a match that will occur at the center of the LDSP. The works done in the field have shown simulation results using DS outperform TSS with a matched NTSS in terms of compensation error. It also has shown a much refined computational cost of the order of 20 to 25 percent. DS has shown better prediction quality and lower complexity than TSS and 2D log search, making it a more preferred choice for software implementations. For hardware implementation, the two diamond sizes, however, makes the control circuitry slightly more complex. Another noteworthy work to address this issue with an end application of stabilization has been done by Song, Ma et al[16] wherein they have incorporated Hooke Jeeves algorithm into the traditional diamond search based fast block matching method, which as per their results has resulted in the efficiency of motion estimation being significantly improved. The testing has been done with 36 blocks (8×8 pixels) is selected in each frame with the proposed method consumed 24.79ms while 30.99ms for DS, the average time processing the same frame of 10 consecutive frames; For a sample set of 49 blocks, it again faired better with 25.39ms verses 31.79ms. When the blocks are increased to 64, the proposed method (35.47ms) rather than DS (41.35ms), still being a real time video stabilization method.

2.4. Feature Matching Methods

Feature matching works on identifying scenes that are easily recognizable. Here, motion estimation can be performed by computing the displacements of these points of interest in the entire video frame by frame and by tracking the positions of these points of interest using the properties of the selected features, forming trajectories. One significant advantage is the fact that the same point can be tracked and recognized across many frames. Some of the more commonly used features detection algorithms are the Kanade Lucas Tomasi (KLT) feature tracker [52] that has been used to track features across videos in several methods [8,16, 20,21,32,34].

Harris corners operator is also a features detector used for this purpose [53]. These features are then tracked using optical flow technique.

SIFT based mechanisms have been widely used [54,38,40,41,43,45,33]. These features use descriptors based on the image gradient to obtain very specific descriptors with very reliable matching results. The descriptors are rotation invariant, but the fact that, they are slower than most alternatives, which makes them too complicated for realtime applications. SURF points although designed on similar principles [35], but are optimized for speed, making them a good alternative [46,8].

Other interesting features like Maximally Stable Extremal Regions (MSER) [55] or FAST corners using BRIEF descriptors [56,26] have been successfully used. Feature matching provides accurate and fast results, and the obtained trajectories allow for additional temporal analysis in the remaining steps of the process, although scenes with large uniform regions can sometimes yield few features per frame. This is one of the limitations of this kind of feature matching methods.

3. Real time Embedded Video Stabilization Systems used for different applications

This section primarily focuses on the works that have been done keeping in mind the realtime application of the end product and have dealt with a hardware implementation of the proposed algorithm. FPGA based prototyped implementations can give a correct analysis of the real-time usability of the VS algorithm as we can analyze the algorithms for computational intensiveness, resource utilization, speed, accuracy and output quality when implemented in a prototyping platform. Most of these works have been targeted for end applications like UAVS, off road vehicles, Robots etc, requiring different approaches, making use of varied computational resources for them to be implementable on embedded systems in real time. However, most of the referred methods have involved finding the 2D motion model to estimate the global motion path. Then the path is low pass filtered to remove the high frequency jitter component. The low frequency parameters are then posted onto frames. This mechanism has been found to be useful for sequences with minimal dynamic movement which is critical to for stabilizing UAV aerial videos. However, it becomes complicated when immediate stabilization, is required like in realtime processing only presuming the global motion path using a given window of frames.

In [26], Oriented FAST and Rotated BRIEF algorithm was utilized for feature extraction followed by matching between consecutive frames. Here, interframe motion was computed using using Random Sample Consensus (RANSAC). The authors proposed framework to implement the whole feature based video stabilization process on a single FPGA chip, in order to achieve real time performance. A fully pipelined FPGA architecture was proposed to substantially accelerate feature based video stabilization in a highly parallel manner, which also provides a reference to accelerating other feature-based video processing tasks such as object tracking and video fusion on FPGA. The authors in [27] worked on a design specifically designed for real time embedded systems where the present and previous video frames were used to estimate the motion vector, followed by filtering operations. The implemention was done on a a Xilinx Spartan 6 LX45 board that was tested on a 640x480 pixel IR camera output. In Vazquez and Chang [28], a smoothening operation was done using the Kanade Lucas tracker to enable the detection of interest points. The undesired motion was balanced out by adjusting for additional rotation and displacements due to the vibrations. The stabilizing speeds achieved through this mechanism have been found to be in the range of 20fps to 28fps for images having resolutions roughly of the order of 320x240 pixels. The proposed system was finally tested on MAC using a 2.16GHz Processor clock and a three frame delay.

Wang [29], suggested a video stabilization method specifically designed for UAVs. In this literature, he proposed a 3 step mechanism using a FAST corner detector to locate the feature points existing in frames. In the second step, after matching, the key points were then used for estimation of affine transform to separate false matches. [30] discussed the video stabilization that has been implemented on an FPGA in the form of a Driven Mobile Robot. Further works include [31] where a real time video stabilization system has

been implemented using high frame rate jitter sensor and a HS camera to retrieve feature points in gray level 512×496 image sequences at 1000 fps. In (32), the stabilization work is based on local trajectories and robust mesh transformation. Every frame is refined in a different local trajectory matrix, helpful in modeling a nonlinear video stream. The method has been proposed to be performing better than conventional methods based on feature trajectories. L. M. Abdullah, N. Md Tahir et. al[36] use a system where Corner Detector System Object is used to find corner values using Harris Corner Detection which is one of the fastest algorithms to find corner values. After the salient points from each frame are obtained the correspondence between the points that are identified previously need to be picked. Authors in [33] have presented a feature-based approach for video stabilization that produces stabilized videos, while preserving the original resolution. . Lowe's method is used to extract extract the SIFT feature points for every frame in a video. Fischler and Bolles's RANSAC algorithm is used in the final fine stage to screen out the feature pairs of mismatch. Then, top eight pairs of feature points are chosen to form a linear system for deriving the perspective transformation between two consecutive frames. Another work that falls in line with the real time implementation of Video Stabilization in hardware has been represented in [34] Here an embedded real time video stabilization system targeted toward a Virtex-5 FPGA Xilinx board is given. The horizontal and vertical global video frame movements are calculated by making use of an integral projection matching approach

4. Advantages of using Feature based extractors like SIFT and SURF for real time applications

Apart from the other descriptors, the focus has been primarily on the works using SIFT and SURF descriptors, especially when it comes to real time stabilization works. It is because of the fact that their accuracy has been pretty good, but more so because they are computationally efficient. Scale invariant feature transform (SIFT) extracts and connects feature points in images which are invariant to image scale, rotation and changes in illumination. Moreover, it provides distinctive descriptors that can find the correspondences between features in different images. Because of all these advantages, it is very suitable for estimating motion between images and hence can be very effectively deployed for motion estimation in Video Stabilization.

However, again as per the referred literatures it was inferred that although SIFT has achieved remarkably success in video stabilization, it suffers from comparatively costly computation, especially for realtime applications. This leads to need for viable replacements with a much lower computational cost. Demonstrably, one of the best of these methods that has been found in comparison to other descriptors has been found to be Speeded up robust features (SURF).

SURF deploys the Hessian matrix due to the very reason that it has had a good history of performance on the performing standards of computation time and accuracy. In place of employing a different parameter for choosing the location and the Hessian Laplace detector, Speeded up Robust Features makes use of the determinant of the Hessian matrix for both the operations. For a particular pixel, the Hessian matrix of this pixel can be given as:

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x\,\partial y} \\ \frac{\partial^2 f}{\partial x\,\partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (4)$$

For the purpose of adaptive scaling, the image is then filtered making use of a Gaussian kernel, so that for a point X = (x, y), the Hessian matrix becomes defined as:

$$H((x,\sigma)) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xx}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \quad (5)$$

where Lxx depicts the convolution of the Gaussian 2nd order derivative of the image I at point x, and likewise Lxy and Lyy. Gaussians derivatives are in general optimized for scale space analysis but at the same time without discretization and cropping. Due to this there is significant observed loss in repeatability while image rotations around in odd multiples. This is a major problem when it comes to Hessian based detectors.

Despite this, the detectors have still been found to perform well, and this performance glitch doesn't outclass the advantage of fast convolutions brought as a result of the discretization and cropping. For determinant of the Hessian matrix calculation, convolving is done with Gaussian kernel second order derivative. Lowe used LoG approximations with considerable success. SURF pushed the boundaries of the approximation for both convolution as well as second order derivative even more with the use of box filters. The approximated second order Gaussian derivatives are evaluated at a very low computational cost making use of integral images being independent of size hence enabling SURF algorithm the capability of fast computation which makes it the perfect choice for the applications of this work.

The 9 × 9 box filters as approximations for Gaussian second order derivatives with σ= 1.2 and depicts the lowest scale and the highest spatial resolution for computing the blob response maps. The approximations have been denoted by D_{xx}, D_{yy}, and D_{xy}. representing scale for computing the blob response. The weights applicable to rectangular regions are simple in nature to minimize computation. This finally produces the approximated determinant of the Hessian as:

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2$$

$$w = 0.9(\text{Bays Suggestion})$$

Or

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (6)$$

Making use of box filters and integral images, SURF doesn't require to recursively apply similar filter to the output of a prior filtered layer however can apply such filters of various sizes at similar speeds directly on the actual image, simultaneously. Therefore, the scale space is verified by up scaling the filter size (9×9 to 15×15 to 21×21 to 27×27 etc) instead of recursively minimising the image size. An octave is a series of filter maps achieved by convolving input image with a filter of increasing size, consists of a scaling factor of 2 and also subdivided into a constant number of scale levels. Due to the discrete nature of integral images, the minimum scale difference between 2 subsequent scales depends on the length l0 of the positive or negative lobes of the partial second order derivative in the direction of derivation (x or y), which has been fixed to a third of the filter size length. For the 9×9 filter, this length is 3. For two successive levels, size is incremented by a minimum of 2 pixels in order to keep the size uneven and ensuring presence of the central pixel, finally resulting in an increased mask size. For localizing image interest points and over scales, a non maximum suppression in a 3 × 3 × 3 neighborhood is applied. The scale space is verified by upscaling the filter size in place of recursively minimising the image size. The following layers are then obtained by gradually filtering the image with larger masks. The biggest advantage of this type of sampling is its computational efficiency. Also, as we do not have to downsample the image, there is no aliasing. On the flipside, box filters tend to protect HF components are lost in magnified scenes, thereby limiting scale invariance. SURF descriptor creation is done normally as a two step mechanism, where first step is to fix a orientation on the basis of information from a circular region around the keypoint and then a region is constructed and aligned to the chosen orientation to extract the SURF descriptor. For rotational invariance, SURF identifies an orientation for the interest points. For achieving, SURF involves estimation of the response of Haar wavelets in xy directions. Besides this, the sampling step is totally scale dependent and is finally chosen to be s, with the wavelet responses being estimated at that current scale s. Similarly, increased scaling of the size of the wavelets is also substantial. Hence for this reason, the integral images are again employed for fast filtering. Then finally, the sum of vertical and horizontal wavelet responses is estimated in a scanning area, leading to change in the scanning orientation, and then again recalculating, till the time orientation with largest sum value is found, which in turn is the main orientation of feature descriptor. However, this step employs extra load of computation on the resources especially which are available on applications like offroad vehicles and UAVs in certain cases. Hence in order to further optimizing the SURF algorithm and in order to reduce the complexity for realtime applications, a curtailed version of SURF termed as USURF can be employed. The authors in [35] proposed an upright version of the descriptor that is image rotation invariant resulting in faster computation and better suited for applications where the camera remains more or less horizontal Hence, the modified SURF also called U-SURF (or Upright SURF) has been found to be suited better for realtime applications for discussed applications.

5. Conclusions

Motion estimation and Digital Video Stabilization has been achieved using multiple mechanisms. However, for Realtime performance, considerable success has been performance enhancement has been achieved with the use of Feature based motion estimation mechanisms like SURF. We have seen that systems, and more specifically embedded systems, have shown better performance in terms of better outputs, frame processing speeds and utilization of resources.

SURF as a descriptor has shown a good tradeoff between accuracy and speed. Most of the other descriptors including SIFT had better accuracies than SURF , however still we preferred SURF over the others because of the very fact speed was more important for us to ensure realtime functionality. An even faster variant of SURF i.e U-SURF can be made use of to further enhance the speeds taking into the account the fact that most of the application discussed in this paper have more significant translational motions than rotational motions. The architecture of the real time embedded system can be further optimized for even quicker performance by using pipelining of different stabilization system blocks and can facilitate the input frames to be fed and processed block by block.

References

[1] Mostafa Attaran , Dawood Seyed Javan "Motion Estimation Using the Gradient Method by Genetic Algorithm" *IEEE International Conference on Computer Modelling and Simulation*, 2010,

[2] Peter O'Donovan "Optical Flow: Techniques and Applications" 20053. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. *Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York*, 1995

[3] Z.E Baarir, F. Charif "Fast modified Horn & Schunck method for the estimation of optical flow fields" *IEEE Workshop on Signal Processing Systems (SiPS)*, 2011

[4] T. Kondo, P. Boonsieng and W. Kongprawechnon, "Improved gradient-based methods for motion estimation in image sequences," *2008 SICE Annual Conference, Tokyo*, 2008, pp. 1120-1123, doi: 10.1109/SICE.2008.4654826.

[5] Boonsieng, Pramuk & Kondo, Toshiaki & Kongprawechnon, Waree. (2014). A robust optical flow estimation technique using gradient orientations. *Science Asia. 40. 73. 10.2306/scienceasia1513-1874.2014.40.073, 2014*

[6] W. Xu, X. Lai, D. Xu, N. A. Tsoligkas, "An Integrated New Scheme for Digital Video Stabilization", *Advances in Multimedia, vol. 2013*, Article ID 651650, 8 pages, 2013. https://doi.org/10.1155/2013/651650

[7] A. N. Netravali and J. D. Robbins, "Motion compensated television coding: Part I", Bell Syst. *Tech. J., vol. 58, pp. 631-670, Mar. 1979*

[8] W. Hassen and H. Amiri, "Block Matching Algorithms for motion estimation," *2013 7th IEEE International Conference on e-Learning in Industrial Electronics (ICELIE), Vienna, 2013*, pp. 136-139, doi: 10.1109/ICELIE.2013.6701287.

[9] H. A. Choudhury and M. Saikia, "Survey on block matching algorithms for motion estimation," 2014 *International Conference on Communication and Signal Processing, Melmaruvathur, 2014*, pp. 036-040, doi: 10.1109/ICCSP.2014.6949794

[10] H. Chen, J. Ding and Y. Lee, "Novel Adaptive Rood Path Searches with Small Motion Prejudgments for Fast Block Motion Estimation," 2018 *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu, 2018*, pp. 560-563, doi: 10.1109/APCCAS.2018.8605600.

[11] W. Lei, M. Shi and G. Wang, "Research and Verilog HDL implementation of technology for motion estimation based on new three — step method," 2017 *IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, 2017*, pp. 1715-1719, doi: 10.1109/ITNEC.2017.8285089.

[12] L. Garlin Delphina and VPS Naidu, "Implementation and Validation of Video Stabilization using Simulink" *In: NACES-11, 21-23 Sep. 2011, MSRIT, Bangalore,2011*

[13] Y. Wang, Q. Huang, D. Zhang and Y. Chen, "Digital Video Stabilization Based on Block Motion Estimation," in *2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, 2017* pp. 894-897. doi: 10.1109/ICCTEC.2017.00198

[14] I. Ali, G. Raja, M. Muzammil and A. K. Khan, "Adaptive Modified Hexagon Based Search Motion Estimation algorithm," *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin), Berlin, 2014*, pp. 147-148, doi: 10.1109/ICCE-Berlin.2014.7034206

[15] H. Chen, J. Ding and Y. Lee, "Novel Adaptive Rood Path Searches with Small Motion Prejudgments for Fast Block Motion Estimation," 2018 *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu, 2018*, pp. 560-563, doi: 10.1109/APCCAS.2018.8605600.

[16] Jifei Song, Xiaohong Ma , "A novel real-time digital video stabilization algorithm based on the improved diamond search and modified Kalman filter," 2015 *IEEE 7th International Conference on Awareness Science and Technology (iCAST), Qinhuangdao, 2015*, pp. 91-95, doi: 10.1109/ICAwST.2015.7314026.

[17] Mingzhou Song and Jing-ao Sun, "Motion Estimation in DCT Domain", *IEEE Transactions on Communication, Vol. 25, pp. 1004 - 1009, Aug. 2003*

[18] Y. Keller, A. Averbuch, and M. Israeli, "Pseudopolarbased estimation of large translations, rotations and scalings in images," *IEEE Trans. Image Processing, vol. 14, no. 1, pp. 12–22, 2005*.

[19] Hanzhou Liu, Baolong Guo and Zongzhe Feng, "Pseudo-log-polar Fourier transform for image registration," in *IEEE Signal Processing Letters, vol. 13, no. 1, pp. 17-20, Jan. 2006*, doi: 10.1109/LSP.2005.860549

[20] A. Briassouli and N. Ahuja, "Integration of frequency and space for multiple motion estimation and shape-independent object segmentation," in *IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 5, pp. 657-669, May 2008*, doi: 10.1109/TCSVT.2008.918799

[21] A. Briassouli and N. Ahuja, "Integration of frequency and space for multiple motion estimation and shape-independent object segmentation," in *IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 5, pp. 657-669, May 2008*, doi: 10.1109/TCSVT.2008.918799.

[22] Y. Wang, Q. Huang, D. Zhang and Y. Chen, "Digital Video Stabilization Based on Block Motion Estimation," *2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China*, 2017, pp. 894-897, doi: 10.1109/ICCTEC.2017.00198.

[23] 23.Babagholami-Mohamadabadi B., Jourabloo A., Manzuri-Shalmani M.T. (2012) A Robust Global Motion Estimation for Digital Video Stabilization. In: Thielscher M., Zhang D. (eds) AI 2012: *Advances in Artificial Intelligence. AI 2012. Lecture Notes in Computer Science, vol 7691. Springer, Berlin, Heidelberg*. https://doi.org/10.1007/978-3-642-35101-3_12

[24] Shen, Yao & Guturu, Parthasarathy & Damarla, Thyagaraju & Buckles, Bill & Namuduri, Kamesh. (2009). Video Stabilization Using Principal Component Analysis and Scale Invariant Feature Transform in Particle Filter Framework. *IEEE Transactions on Consumer Electronics, 55(3)*. 1714 - 1721. 10.1109/TCE.2009.5278047.

[25] Chenguang, Guo & Xianglong, Li & Linfeng, Zhong & Xiang, Luo. (2009). A Fast and Accurate Corner Detector Based on Harris Algorithm. *3rd International Symposium on Intelligent Information Technology Application, IITA 2009.* 2. 10.1109/IITA.2009.311.

[26] J. Li, T. Xu and K. Zhang, "Real Time Feature-Based Video Stabilization on FPGA," in *IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 4, pp. 907-919, April 2017*, doi: 10.1109/TCSVT.2016.2515238.

[27] L. Araneda and M. Figueroa, "Real-Time Digital Video Stabilization on an FPGA," 2014 *17th Euromicro Conference on Digital System Design, Verona, 2014*, pp. 90-97, doi: 10.1109/DSD.2014.26

[28] Vazquez, M., Chang, C.: Real-time video smoothing for small rc helicopters. In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, IEEE, pp. 4019–4024 (2009)

[29] Wang Y, Hou Z, Leman K, Chang R (2011) Real-time video stabilization for unmanned aerial vehicles. *IAPR Conference on Machine Vision Applications, June 13-15, 2011, Nara, JAPAN, 2011* pp 336–339

[30] J. Li, T. Xu and K. Zhang, "Real-Time Feature-Based Video Stabilization on FPGA," in *IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 4, pp. 907-919, April 2017*, doi: 10.1109/TCSVT.2016.2515238

[31] Raut, S., Shimasaki, K., Singh, S. et al. Real-time high-resolution video stabilization using high-frame-rate jitter sensing. *Robomech J 6, 16 (2019).* https://doi.org/10.1186/s40648-019-0144-z

[32] Z. Zhao and X. Ma, "Video stabilization based on local trajectories and robust mesh transformation," 2016 *IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, 2016*, pp. 4092-4096, doi: 10.1109/ICIP.2016.7533129.

[33] Y. H. Chen, H. Y. S. Lin and C. W. Su, "Full-Frame Video Stabilization via SIFT Feature Matching," 2014 *Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, 2014*, pp. 361-364, doi: 10.1109/IIH- MSP.2014.96.

[34] J. Li, T. Xu and K. Zhang, "Real-Time Feature-Based Video Stabilization on FPGA," in *IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 4, pp. 907-919, April 2017*, doi: 10.1109/TCSVT.2016.2515238

[35] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "Speeded-Up Robust Features (SURF)", *Computer Vision and Image Understanding, Volume 110, Issue 3, 2008*, Pages 346-359, ISSN 1077-3142, https://doi.org/10.1016/j.cviu.2007.09.014

[36] L. M. Abdullah, N. Md Tahir and M. Samad, "Video stabilization based on point feature matching technique," 2012 *IEEE Control and System Graduate Research Colloquium, Shah Alam, Selangor, 2012*, pp. 303-307, doi: 10.1109/ICSGRC.2012.6287181.

[37] W. -x. Jin, X. -g. Di and S. -w. Fu, "A fast digital image stabilization based on large-angle rotational motion estimation," *Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, 2014*, pp. 5615-5620, doi: 10.1109/WCICA.2014.7053676

[38] Chang Li and Yangke Liu, "Global motion estimation based on SIFT feature match for digital image stabilization," *Proceedings of 2011 International Conference on Computer Science and Network Technology, Harbin, 2011*, pp. 2264-2267, doi: 10.1109/ICCSNT.2011.6182425.

[39] Lim, A., Ramesh, B., Yang, Y. et al. Real-time optical flow-based video stabilization for unmanned aerial vehicles. *J Real-Time Image Proc 16, 1975–1985 (2019).* https://doi.org/10.1007/s11554-017-0699-y

[40] M. Fang, H. Li and S. Si, "A video stabilization algorithm based on affine SIFT," 2018 *International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, 2018*, pp. 1-4, doi: 10.1109/ICOMET.2018.8346332.

[41] M. Sharif, S. Khan, T. Saba, M. Raza and A. Rehman, "Improved Video Stabilization using SIFT-Log Polar Technique for Unmanned Aerial Vehicles," 2019 *International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 2019*, pp. 1-7, doi: 10.1109/ICCISci.2019.8716427.

[42] Huan Yu and Wenhui Zhang, "Moving camera video stabilization based on Kalman filter and least squares fitting," *Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, 2014*, pp. 5956-5961, doi: 10.1109/WCICA.2014.7053740.

[43] Shen, Yao & Guturu, Parthasarathy & Damarla, Thyagaraju & Buckles, Bill & Namuduri, Kamesh. (2009). Video Stabilization Using Principal Component Analysis and Scale Invariant Feature Transform in Particle Filter Framework. *Consumer Electronics, IEEE Transactions on. 55(3)*. 1714 - 1721. 10.1109/TCE.2009.5278047.

[44] Dong, Jing & Liu, HaiBo. (2016). Video Stabilization for Strict Real-Time Applications. *IEEE Transactions on Circuits and Systems for Video Technology.* 27. 1-1. 10.1109/TCSVT.2016.2589860.

[45] Yang, Junlan & Schonfeld, Dan & Mohamed, Magdi. (2009). Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion. *IEEE Transactions on Circuits and Systems for Video Technology.* 19. 945 - 954. 10.1109/TCSVT.2009.2020252.

[46] Gao Chunxian, Zeng Zhe, Liu Hui, "Hybrid Video Stabilization for Mobile Vehicle Detection on SURF in Aerial Surveillance", *Discrete Dynamics in Nature and Society, vol. 2015, Article ID 357191*, 12 pages, 2015. https://doi.org/10.1155/2015/357191

[47] N. Ejaz, W. Kim, S. I. Kwon and S. W. Baik, "Video Stabilization by Detecting Intentional and Unintentional Camera Motions," 2012 *Third International Conference on Intelligent Systems Modelling and Simulation, Kota Kinabalu, 2012*, pp. 312-316, doi: 10.1109/ISMS.2012.73.

[48] Chao Zhanga, Fugen Zhoua, Bindang Xuea, Wenfang Xueb, "Stabilization of atmospheric turbulence-distorted video containing moving objects using the monogenic signal *"Journal of Signal Processing: Image Communication, Elsevier*, Pages 19-29, Volume 63, April 2018.

[49] Xie, Q. & Chen, Xi & Zhang, L. & Jiang, A. & Cui, F. (2016). A novel rapid and efficient video stabilization algorithm for mobile platforms. 1-4. 10.1109/VCIP.2016.7805487.

[50] C. Fang, T. Tsai and C. Chang, "Video stabilization with local rotational motion model," *2012 IEEE Asia Pacific Conference on Circuits and Systems, Kaohsiung, 2012*, pp. 551-554, doi: 10.1109/APCCAS.2012.6419094.

[51] M. Wang et al, "Deep Online Video Stabilization With Multi-Grid Warping Transformation Learning," in *IEEE Transactions on Image Processing, vol. 28, no. 5*, pp. 2283-2292, May 2019, doi: 10.1109/TIP.2018.2884280.

[52] Carlo Tomasi, Takeo Kanade, Detection and Tracking of Point Features, School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[53] Javier Sánchez, Comparison of motion smoothing strategies for video stabilization using parametric models, Image Process. Online 7 (2017) 309–346.

[54] Bing-Yu Chen, Ken-Yi Lee, Wei-Ting Huang, Jong-Shan Lin, Capturing intention based full-frame video stabilization, Comput. Graph. Forum 27 (7) (2008) 1805–1814.

[55] Manish Okade, Prabir Kumar Biswas, Video stabilization using maximally stable extremal region features, Multimedia Tools Appl. 68 (3) (2014) 947–968.

[56] Semi Jeon, Inhye Yoon, Jinbeum Jang, Seungji Yang, Jisung Kim, Joonki Paik, Robust video stabilization using particle keypoint update and l1-optimize camera path, in: Sensors, Sensors 17 (2) (2017) 337.

Table 1 Comparative analysis of Motion estimation Techniques and the relevant work in DVS based Stabilization

Author(s)	Motion Estimation Method	Motion Model	USP	Limitations
Jifei Song ,X. Ma[16]	Block Based(Diamond Search)	2D	Incorporating Hooke Jeeves into traditional DS	Makes use of Bilinear interpolation, thereby making it time consuming and slow
Y. Wang, Q. Huang[13]	Block Motion Estimation	2D	Use of SAD in place of MSE, SSD reduces computational intensity.	Reduced picture resolution due to Low pass filtering , No performance metrics evaluated
W. -x. Jin, X. -g. Di[37]	Block Motion Estimation	2D	Bit Plane Matching, Performs well even for large angle rotational motion, Speeds upto 29 fps	Estimation of translational after rotational motion makes it complicated
Chang Li and Yangke Liu [38]	Feature Based	2D	Block based key point extraction for SIFT to reduce complexity	Decreased Accuracy due to use of blocks, Below average performance for scenes with moving scenes

Lim, A., Ramesh, B., Yang[39]	Feature Based	2D	Processing Speed of 50 fps, Truly realtime, Multithreaded Work Flow	Performance drops in scenes with constant motion
J. Li, T. Xu and K. Zhang[26]	Feature Based (Oriented FAST and Rotated BRIEF and RANSAC integrated)	2D	FPGA based implementation, Pipelined Architecture	Stringent hardware requirements for embedded Processors
M. Fang, H. Li and S. Si[40]	Feature Based(Affine SIFT)	2D	Complete Affine Variance as a result of usage of ASIFT	Use of Affine SIFT and Matsushita Gaussian Filter makes it non real time.
M. Sharif, S. Khan[41]	Feature Based (SIFT with Log Transform	2D	Rotation invariance	Requires Storing of Video Sequence, Not realtime
Babagholami-Mohamadabadi B., Jourabloo[23]	Frequency Transform based(2D Radon transform, 1D Fourier transform and 1D Scale transform)	2D	Rotational Invariance upto +-15 degrees	Doesn't deal with affine models, Frequency domain issues render it useless for real time
Huan Yu and Wenhui Zhang[42]	Global Motion estimation, Corner matching by DS Method	2D	Enhanced Kalman filtering for motion correction stage	Diamond search for corner matching, complicated kalman filtering stage
Yao Shen, Parthasarathy Guturu[43]	Feature Based (PCA -SIFT)	2D	Use of the SIFT-BMSE for particle weight determination further improves its speed and accuracy	Requires adaptively changing particle size to reduce the complexity
W. Xu,X. Lai[6]	Gradient Descent(Optical Flow motion estimation technique)	2D	Use of Savitzky-Golay filter in place of Kalman filters for processing estimated affine global motion parameters,	Not realtime
L. Garlin Delphina and VPS Naidu[12]	Block Based Motion Estimation (TSS along with GCBP)	2D	Good PSNR and ITF values for Bloack based mechanisms	GCBP Computation is slow and also assumes within 4 chosen sub images as uniform
Jing Dong, Haibo Liu[44]	Homography Estimation	2D	Truly realtime performance	Homographies are employed to represent the video motion over a short time interval only. Hence assumes virtually no camera translation within a short timeframe
Yang, Junlan & Schonfeld[45]	SIFT	2D	Particle Filtering for Global Motion Estimation Between Successive Frames	Use of SIFT reduces realtime capabilities
Wang Y, Hou Z, Leman K[29]	FAST Corner Detection	2D	Realtime	Max performance till 30 fps, keypoint matching is based on grey image only
Gao Chunxian, Zeng Zhe[46]	SURF	2D	Tradeoff between accuracy and Speed	Computational overhead due to lack of pipelining and multithreading in design architecture
Raut, S., Shimasaki[31]		2D	Completely Realtime, Works on HR Images	For Grayscale images
N. Ejaz, W. Kim [47]	Optical Flow based motion esti mation	2D	Use of Image morphing to reconstruct jittered frames	Low PSNR values
Chao Zhanga, Fugen Zhoua[48]	Frequency Based	2D	Generating stable background frames using the monogenic signal	Not realtime due to presence of complex transforms
Z. Zhao and X. Ma[32]	Feature points extraction (KLT)	2D	Feature points by Kanade Lucas Tracker and Outlier removal by RANSAC	Very low processing speed of 8 fps

Xie, Q. & Chen[49]	Global motion estimation(K Means Clustering)	2D	Applicable to handheld devices with optimal resources	Coarse to fine global motion estimation method based on the tiny block matching suffers from computation cost and complex motion vector classification
A. J. Amiri and H. Moradi[8]	SURF	2D	Use of BRISK as detectors along with RANSAC	15 fps only
C. Fang, T. Tsai[50]	Block Matching Estimation	2D	Use of rotation-based block matching method for local motion estimation	Not realtime
Y. H. Chen, H. Y. S. Lin[33]	SIFT	2D	Full frame Stabilization	Not realtime
M. Wang[51]	Deep Neural Networks	2D	Development of StabNet Database	Frames need to be buffered and stored
L. Araneda and M. Figueroa[27]	Global Motion Estimation	2D	Fast and realtime	Better MSE results can be obtained