Implementation and Validation of Video Stabilization using Simulink

L. Garlin Delphina and VPS Naidu
MSDF Lab, FMCD
CSIR - National Aerospace Laboratories
Bangalore, India
vpsnaidu@gmail.com

Abstract— **A fast video stabilization technique based on Gray-coded bit-plane (GCBP) matching for translational motion is implemented and tested using various image sequences. This technique performs motion estimation using GCBP of image sequences which greatly reduces the computational load. In order to further improve computational efficiency, the three-step search (TSS) is used along with GCBP matching to perform a competent search during correlation measure calculation. The entire technique has been implemented in Simulink to perform in real-time.**

Keywords- Real-time Video stabilization, Gray-coded Bit Plane Matching, Three Step Search algorithm

I. Introduction

Unmanned Aerial Vehicles carry visual and infrared cameras to capture outside scenes in real-time. The usefulness of the imagery that these flying machines relay to the operators in military applications is directly related to how much of information they can 'extract' from the frames being captured in real-time. Generally the captured video looks jittery and not appealing due to vibrations of the platform. To address this issue of oscillations and low frequency vibrations in the recorded imagery, implementation of a Digital Image Stabilization (DIS) algorithm is required to improve visual quality. Furthermore, the stabilization algorithm needs to be computationally inexpensive to perform in real time. Hence, implementation of GCBP matching using TSS Algorithm [1,2] using Simulink has been illustrated in this paper. It reduces the computational overhead by realizing the stabilization technique in real-time using binary Boolean functions.

II. Video Stabilization

The top level implementation of DIS based on GCBP matching using TSS in Simulink is shown in Fig.1. It consists of the following blocks [3,4]:

- Video reading and grayscale conversion
- Motion estimation
  - Bit plane decomposition
  - Local motion estimation
  - Global motion estimation
  - Integrated motion estimation
- Motion compensation
- Peak Signal to Noise Ratio (PSNR) and Inter-frame Transformation Fidelity (ITF) estimation

A. *Video reading and grayscale conversion*

An input video is read frame-by-frame in real time at a desired sampling rate of 15 frames per second. The output color format is selected to be 'RGB' of class 'uint8'. An output end-of-file (EOF) indicator is then used to determine when the last video frame was output from the block. When a frame read is the last frame of that video, the output from EOF port becomes 1. Otherwise, the output from the EOF port remains 0. A STOP block stops the simulation when input i.e. EOF port is nonzero.

B. *Motion estimation*

The Motion estimation subsystem consists of the following internal blocks as shown in Fig.2:

B1. *Bit plane decomposition*

The image data is represented in gray values and is decomposed into eight gray coded bit planes (GCBP) i.e. (K = 8) using standard binary representation. For t^{th} image frame with 2^K gray levels, the gray level value $f^t(x,y)$ at the pixel location (x,y) can be represented as:

$$f^t(x,y) = a_{k-1}2^{k-1} + a_{k-2}2^{k-2} + ... + a_0 2^0 \qquad (1)$$

where, a_k is the binary coefficient which can either be 0 or 1 and $0 \le k \le K-1$.

The bit planes of this image are represented by $b_0{}^t(x,y)$ (LSB) to $b_7{}^t(x,y)$ (MSB). This t^{th} image frame is represented by gray bit codes g_7 to g_0 as:

$$g_k = a_k\,, \quad k = K-1 \qquad (2)$$

$$g_k = a_k \oplus a_{k+1}\,, \quad 0 \le k \le K-2 \qquad (3)$$

where $\oplus$ is the exclusive OR operator.

The gray codes for bits other than MSB are obtained by exclusive-OR operation (Eq.3) and the successive code words differ only in one bit position. An optimal gray bit plane (obtained in Section III) is then divided into four sub-images (S1,S2,S3,S4) each of equal size (half the width and height of image frame).

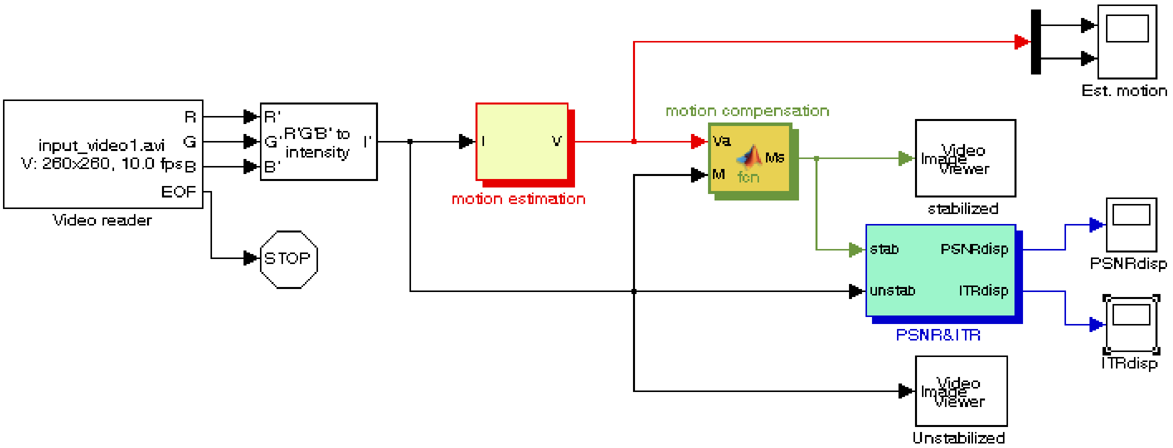

Fig.1. Top level Simulink implementation of Video stabilization

B2. Local motion estimation

Local motion vectors are estimated from the four sub-images placed in appropriate positions in the GCBP. Each motion vector of a sub-image in the current GCBP image is determined by evaluating GCBP matching over sub-images in the previous GCBP and selecting the sub-image which yields the closest matching. This approach assumes that all pixels within the sub-image have uniform motion and the range of motion vector is constrained by the search window. Let the size of each block be $R\times C$ and a search window range from p to $p+N$ in each block, where p is the maximum expected displacement in the search window. N is the matching block size assumed within each block for a faster search as shown in Fig.3. For a GCBP matching, correlation measure is:

$$Corr_j(r,c)=\frac{1}{RC}\sum_{x=0}^{R-1}\sum_{y=0}^{C-1} g_k^{\,t}(x,y)\oplus g_k^{\,t-1}(r+x,c+y) \qquad (4)$$

$$\text{where,}\ \begin{array}{l} -p\le r,c\le p \\ 1\le j\le 4 \end{array}$$

where, $g_k^{\,t}(x,y)$ and $g_k^{\,t-1}(x,y)$ are the current and previous k^{th} order GCBPs. The correlation measure gives the unmatched bits between the current and the previous bit planes of j^{th} sub-block. Therefore, the smallest measure gives the best matching between the bit planes [5]. In order to select a search window centered on each sub-image, the matching block size N can be generalized as follows:

$$N=0.5h-2p \qquad (5)$$

where, h = height of the image in pixels.

Maximum displacement p was assumed to be 9 (by intuition) considering the fact that the pixel displacement between successive frames would not go beyond 9 pixels. A matching block size N = 112 was considered to measure Correlation for a 260x260 image. In order to reduce the computational complexity in calculating correlation measure, GCBP matching is combined with TSS algorithm [6,7]. TSS is a fine-coarse search mechanism as shown in Fig.4. The local motion vector V_j for each j^{th} sub-image is then calculated based on a minimum correlation value obtained from Eq. 4.

B3. Global motion estimation

All the motion vectors V_j are stacked together with the global motion vector $V_g^{\,t-1}$ from the previous frame and passed through a median filter to obtain the current global motion vector $V_g^{\,t}$.

B4. Integrated motion estimation

The global motion vector of a frame is then integrated with a damping coefficient to obtain the integrated motion vector that designates the final motion vector of a frame for motion correction. The integrated motion vector V_a for smooth

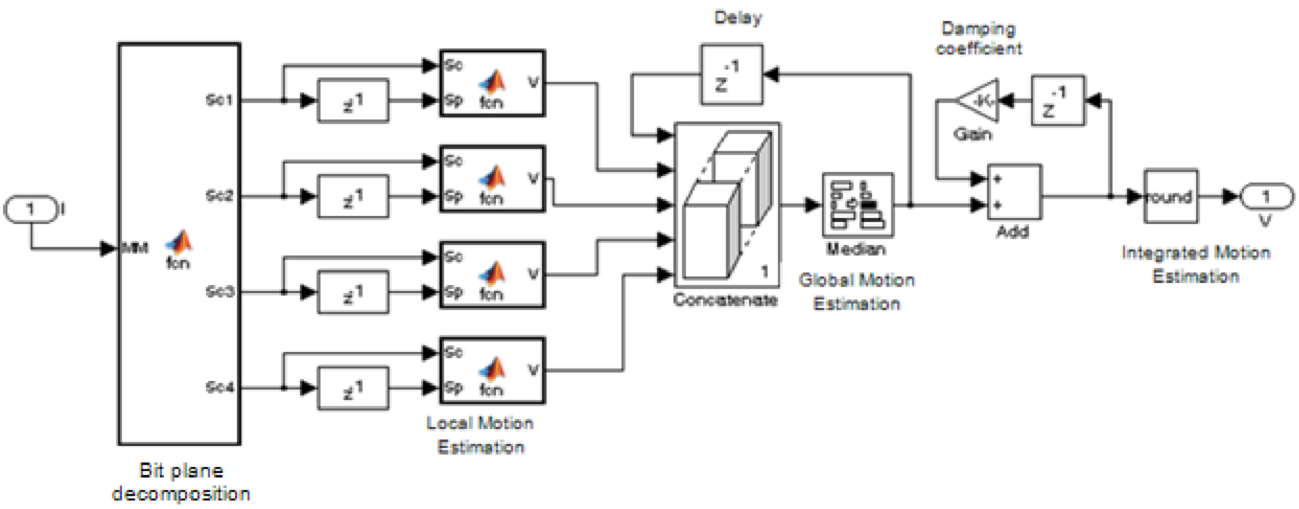

Fig.2. Motion Estimation block

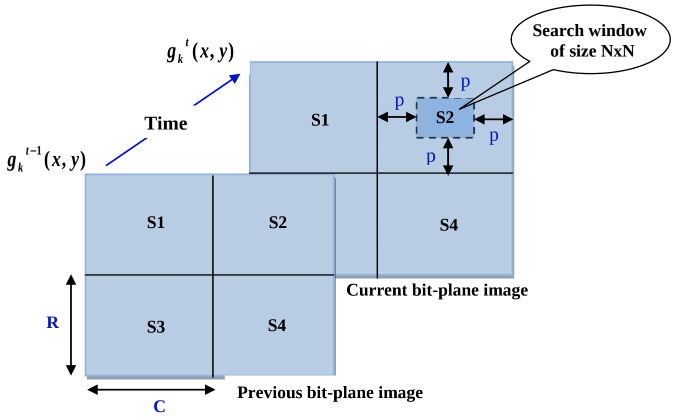

Fig.3: Local motion vector estimation using successive bit plane images

panning is:

$$V_a^{\,t} = D\,V_a^{\,t-1} + V_g^{\,t} \qquad (6)$$

where $V_g^{\,t}$ is a global motion vector and $D \quad (0 < D < 1)$ is a damping coefficient, which enables integrated motion vector converge to zero when there is no camera motion. Damping coefficient is achieved using a gain block of gain 0.95, and the integrated motion vector is rounded to designate the final motion vector of a frame for motion correction.

C. *Motion compensation*

Using the rounded integrated motion vectors as the number of rows and columns to be adjusted, the original image pixels are relocated to remove all the estimated motion, yielding a stabilized image.

D. *PSNR and ITF estimation*

The PSNR and ITF sub-system shown in Fig. 5 consists of PSNR and ITF estimation blocks. PSNR is calculated to evaluate the performance of the motion estimation algorithm by comparing the stabilized video to that of the original destabilized video frame by frame. This ratio is used as a quality measurement between the original and the stabilized videos. The higher the PSNR, the better is the quality of the stabilized image sequence. PSNR is calculated as:

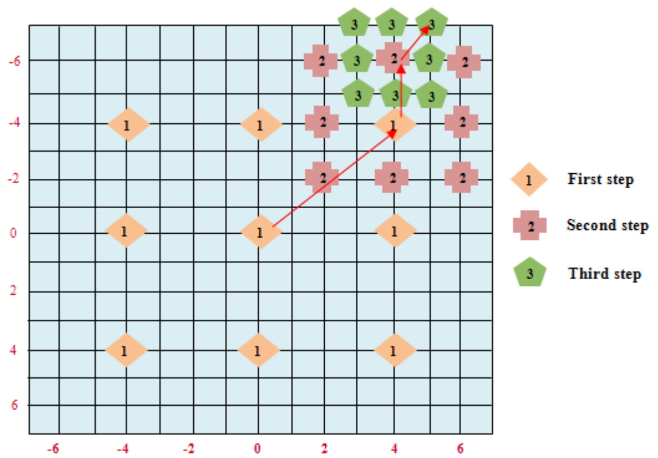

Fig.4: Search pattern for Three Step Search algorithm

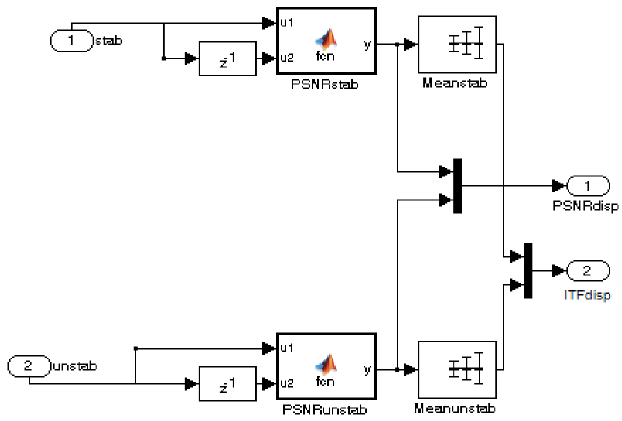

Fig.5: PSNR and ITF sub-system

$$PSNR = 10\,\log_{10} \frac{I_{\max}^2}{\sqrt{\dfrac{1}{MN}\displaystyle\sum_{m=1}^{M}\sum_{n=1}^{N}\left(I_t(m,n) - I_{t+1}(m,n)\right)^2}} \qquad (7)$$

where $I_t(m,n)$ is image at current time instant; $I_{t+1}(m,n)$ is image at next time instant; M, N are the number of rows and columns in the image and $I_{\max}$ is the maximum gray value in the input image i.e. 255. PSNR values of stabilized and destabilized images are thus calculated and fed to the ITF block to estimate mean. These mean values define Inter-frame Transformation Fidelity (ITF) which is expected to be high for stabilized video.

III. Results and Discussions

Two different simulated video sequences are used to test the efficiency of the stabilization algorithm. The first image sequence is simulated using a 512x512 image containing two fighter aircrafts as shown in Fig.6a. The video is simulated by subjecting the image to sine wave oscillations of magnitude 10 pixels in vertical, and -5 pixels in horizontal directions at 10Hz for duration of 25sec. From the centre of the oscillated images, 260x260 pixels are considered to simulate an unstable video as shown in Fig.6b. In order to obtain a single optimal gray coded bit plane, various simulations are performed to analyze the estimated motion vectors obtained for each bit plane as shown in Figs.7 & 8 It can be seen that

a) b)

Fig.6 a) Original 512x512 image b) 260x260 image

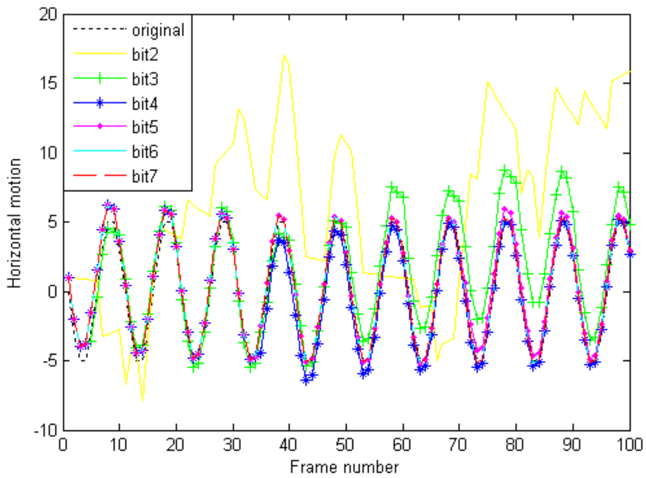

Fig.7: Horizontal motion vectors corresponding to various bit planes

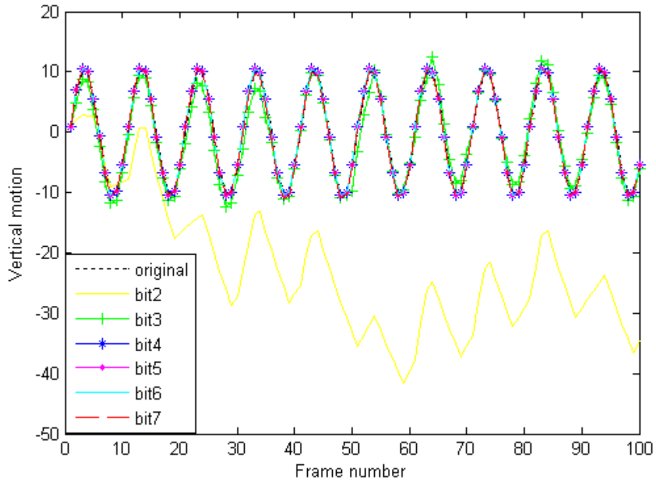

Fig.8: Vertical motion vectors corresponding to various bit planes

the bit planes 2 & 3 fail to appropriately estimate the actual motion, while higher order bit planes 4, 5, 6 & 7 succeed in the same. The PSNR values of stabilized video corresponding to each bit plane, as given in Table.1 also shows that, the gray bit plane obtained by exclusive-OR of higher order bits 6 and 7 can be suitably used for motion estimation. The maximum number of horizontal and vertical motion estimated in pixels at each time instant as shown in Fig.9 indicates that the algorithm has correctly detected the 10 pixel displacement in vertical direction and -5 pixel displacement in horizontal direction used initially for simulating the video. The PSNR and ITF displays as shown in Fig.10 & 11 respectively confirm that the quality of stabilized video is better than that of destabilized video.

The Simulink model is again tested with a different simulated input video sequence to prove the efficacy of the algorithm and the Simulink implementation. A video sequence of 100 frames (10sec) and resolution

TABLE .1: PSNR values considering different bit planes

PSNR (dB)	Bit=2	Bit=3	Bit=4	Bit=5	Bit=6	Bit=7
Destabilized	32.46	32.46	32.46	32.46	32.46	32.46
Stabilized	33.32	38.34	39.84	39.90	**39.94**	39.86

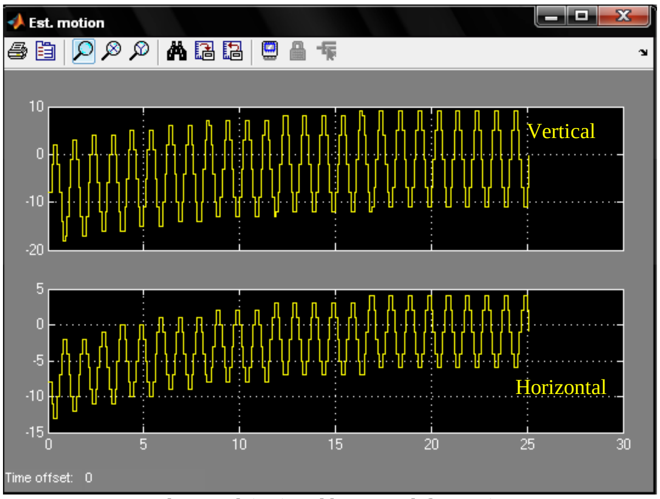

Fig.9: Estimated vertical (top) and horizontal (bottom) motion vectors

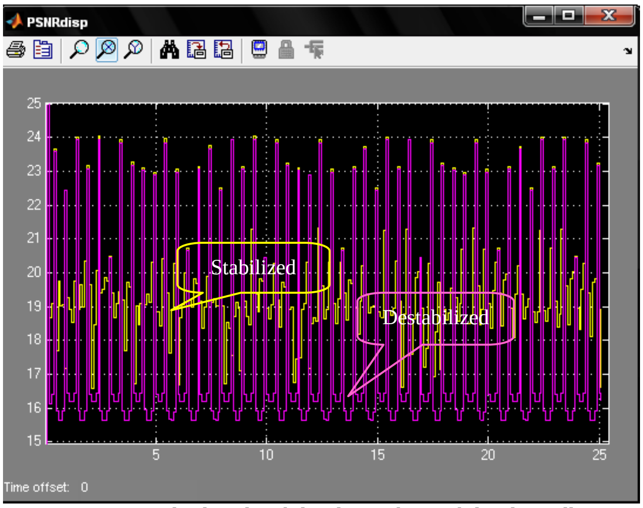

Fig.10: PSNR display: destabilized in pink & stabilized in yellow

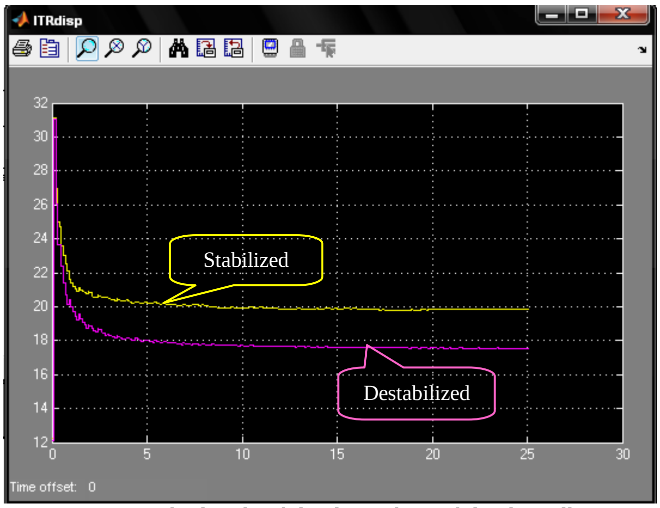

Fig.11: ITF display: destabilized in pink & stabilized in yellow

260x260 is simulated using 100 random numbers. Table.2 containing PSNR values of different bit planes prove that bit plane 6 is optimal for the algorithm.

TABLE .2: PSNR values considering different bit planes

PSNR (dB)	Bit=2	Bit=3	Bit=4	Bit=5	Bit=6	Bit=7
Destabilized	32.18	32.18	32.18	32.18	32.18	32.18
Stabilized	33.42	33.37	38.00	38.83	**38.97**	38.95

The PSNR and ITF display outputs shown in Figs.12 & 13 respectively substantiates the fact that the algorithm has been successful in stabilizing the video effectively.

IV. CONCLUSION

Computationally efficient digital image stabilization (DIS) technique based on Gray coded bit plane matching and three step search algorithm has been implemented and validated successfully in Simulink. The algorithm's efficiency is improved by binary Boolean operations, reduced pixel search and has the competence to be implemented in real-time too, due to its simplicity. Effective motion compensation technique has been achieved using integrated motion vectors enabling unwanted motion control from any number of frames. Simulation and analysis of estimated motion vectors and PSNR values for different bit planes have enabled to choose the best bit plane for the algorithm. Extensive tests using simulated videos confirm the efficacy of the algorithm. Final stabilized video with appropriately estimated motion vectors show that an effective and satisfactory video stabilization with reduced motion between frames has been achieved for translational motion.

The algorithm has been executed for gray images, but it can be extended to color images too by applying the estimated motion vectors of the gray format independently to red, green and blue components for motion compensation. Though a single, optimal gray bit plane has been selected for correlation calculation, it is always advisable to consider all higher order bit planes with good amount of information to estimate motion vectors with high degree of accuracy. In a real time implementation, instead of using a first order filter for smoothing the estimated motion vectors, Kalman filter would be a better choice to reduce noise and also predict motion in the subsequent frames accurately.

REFERENCES

[1] S.J. Ko, S.H. Lee, and K.H. Lee, "Digital image stabilizing algorithms based on bit-plane matching", IEEE Trans. on Consumer Electronics, vol. 44, No. 3, pp, 617-622, Aug. 1998.

[2] S. Ko, S. Lee, S. Jeon, and E. Kang, "Fast digital image stabilizer based on gray-coded bit-plane matching", IEEE Transactions on Consumer Electronics, vol. 45, no. 3, pp. 598-603, Aug. 1999.

[3] Mariano I. Liz´arraga and Sonia Arteaga, "UAV Video Stabilization", CMPE264: Image Analysis and Computer Vision, August 15, 2007.

[4] A. Brooks, "Real-Time Digital Image Stabilization", Image Processing Report, Department of Electrical Engineering, Northwestern University, Evanston, IL, 2003.

[5] J. R. Jain and A. K. Jain, "Displacement measurement and its application in inter-frame image coding," IEEE Trans. on Communications, Vol. com-29, no. 12, pp. 1799-1808, Dec. 1981.

[6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion compensated interframe coding for video conferencing", Pro. Nat. Telecom. Conf., New Orleans, pp. G5.3.1- 5.3.5, Nov. 1981.

[7] R. Li, B. Zeng and M. L. Liou, "A new three step search algorithm for block motion estimation", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 4, No. 4, pp.438-442, Aug. 1994.

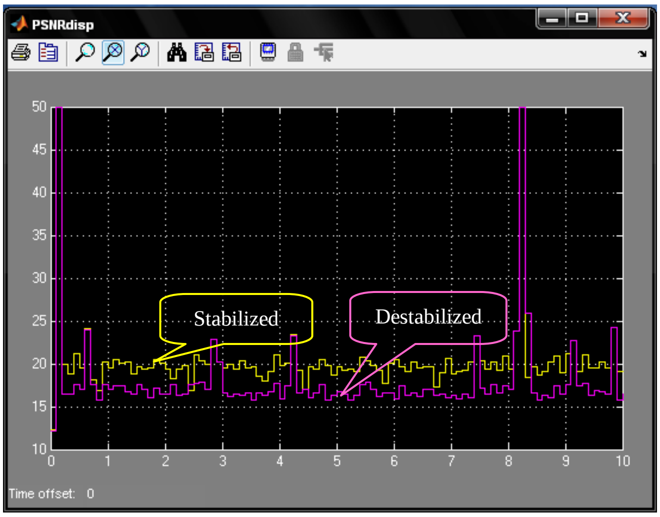

Fig.12: PSNR display: destabilized in pink & stabilized in yellow

Fig.13: ITF display: destabilized in pink & stabilized in yellow