DIGITAL VIDEO STABILIZATION SYSTEM BY ADAPTIVE FUZZY FILTERING

M.J. Tanakian, M. Rezaei, and F. Mohanna

Faculty of Electrical & Computer Engineering, University of Sistan and Baluchestan,
Zahedan, Iran
j.tanakian@gmail.com, mehdi.rezaei@ece.usb.ac.ir, f_mohanna@ece.usb.ac.ir
web: www.usb.ac.ir

ABSTRACT

Digital video stabilization (DVS) allows acquiring video sequences without disturbing jerkiness, removing unwanted camera movements. In this paper, we propose a novel DVS algorithm that adaptively compensates the camera jitters utilizing an adaptive fuzzy filter on the global motion of video frames. The adaptive fuzzy filter is a simple infinite impulse response (IIR) filter which is tuned adaptively by a fuzzy system. The fuzzy system has two inputs which are used as quantitative representations of unwanted and intentional camera motion. The fuzzy system is also tuned adaptively during operation according to the characteristics of camera jitters. The global motion of video frames is estimated based on the block motion vectors which are resulted by video encoder during motion estimation operation. The proposed method also utilizes an adaptive criterion for validating of motion vectors. Experimental results have indicated a good performance for the proposed DVS algorithm.

1. INTRODUCTION

DIGITAL video stabilization techniques have been studied for decades to improve visual quality of image sequences captured by compact and light weight digital video cameras. When such cameras are hand held or mounted on unstable platforms, the captured video generally looks shaky because of undesired camera motions. Unwanted video vibrations would lead to degraded view experience and also greatly affect the performances of applications such as video encoding [1-2] and video surveillance [4]. With recent advances in wireless technology, video stabilization systems are also considered for integration into wireless video communication equipments for stabilization of acquired sequences before transmission, not only to improve visual quality but also to increase the compression performance [1]. A DVS system is implemented by software, which makes it easy to be miniaturized and updated. Consequently, DVS is suitable for portable digital devices, such as digital camera and mobile phone.

Generally a DVS system consists of two principal units: motion estimation (ME) and motion correction (MC) units. The ME unit estimates the global motion vectors (GMVs) between every two consecutive frames of the input video sequence. Using these GMVs, the MC unit then generates the smoothing motion vectors (SMVs) needed to compensate the frame jitters and warp the frames to create a more visual stable image sequence.

In the context of video stabilization, most previous approaches attempt to reduce the computational cost of ME by using fast algorithms [3, 5-7], or by limiting the global ME to small, pre-defined regions [5, 8]. Such approaches consider digital video stabilization and video encoding separately and attempt to trade the accuracy of motion vectors (MVs) for the computational efficiency. Nevertheless, they improve the computational efficiency at the expense of degradation in the accuracy of motion vectors. Since both the video encoder and the digital stabilizer of a digital video camera need to compute the image motion, we can integrate digital stabilizer with video encoder by making the two modules of a digital video camera share a common local motion vectors (LMVs) estimation process, as shown in figure1.

One of the essential tasks in DVS is to separate the unwanted hand jitters from the intentional camera movement utilizing the block motion vectors. The block motion vectors as local motion vectors are used for global camera motion estimation and compensation. Among the various MC algorithms proposed in the literature, smoothing of the global motion vector by low pass filtering is the most popular [9]. Kalman filter and fuzzy systems have widely been used in DVS applications. A membership function adaptive fuzzy filter for image sequence stabilization is presented in [10] and a DVS system consists of a fuzzy system and the Kalman filter is presented in [11].

In the proposed DVS in this paper, the global motion vector is estimated based on the block motion vectors (BMVs) which are estimated by the video encoder. The BMVs are estimated based on block matching using difference criteria such as MAD (Mean Absolute Difference). In a video sequence with smooth or rough regions, the

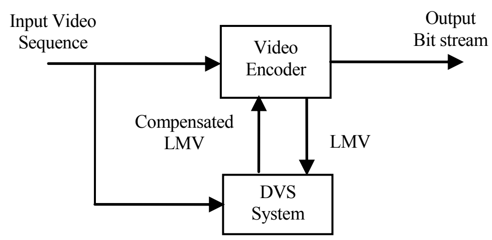

Figure 1- Integration scheme of the video stabilizer and the video encoder

estimated BMVs may not be in coincidence with the real motion of the blocks. Although such LMVs are applicable to the local motion compensation task which is executed in the encoder, they cannot be used for the global motion compensation which is executed by the DVS. These LMVs include some noises that degrade the global motion estimation task. Two qualifying tests, namely “Lack of Features” and “Low SNR”, are used in [9] to remove the noisy LMVs by a simple thresholding algorithm. An adaptive thresholding algorithm is proposed in this paper for removing the noisy LMVs and a global motion vector is computed for each video frame based on the filtered LMVs. Then, a fuzzy adaptive IIR filter is applied to the global motion vectors to smooth the unwanted camera motions. The IIR filter is tuned adaptively by a fuzzy system and the fuzzy system is also tuned adaptively according to the characteristics of camera motions. Experimental results show a good performance for the proposed DVS algorithm.

The rest of this paper is organized as follows. The details of the proposed DVS algorithm are described in Section 2. Some experimental results are presented in Section 3 and the paper is concluded in section 4.

2. THE PROPOSED METHOD

The flowchart of the proposed DVS system is depicted in figure 2. The details of proposed system are described in the sequel.

2.1 Block-Based Motion Estimation

The block-based motion estimation is used to generate the local motion vectors (LMVs). Since the motion estimation is executed by the video encoder, there is no any computational complexity cost for the DVS. In this paper, to test the proposed DVS independent of the encoder, a full search motion estimation algorithm with full-pixel resolution is used for 8×8 blocks with a 33×33 search area.

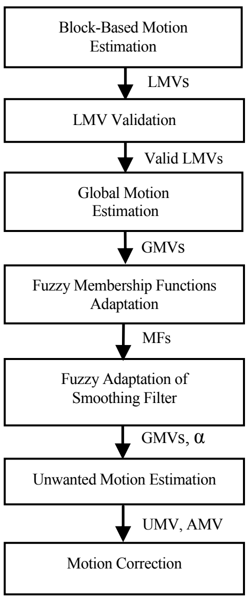

Figure 2- Flowchart of the proposed DVS system.

2.2 LMV Validation

The block motion estimation module typically computes some wrong motion vectors which are not in coincidence to the real motion direction of the blocks. Although, such motion vectors can be used for the block motion compensation and encoding, they include noise and should not be used for the global motion compensation and video stabilization. The noisy LMVs are separated from the valid motion vectors by using an adaptive thresholding method. The noisy motion vectors are mostly obtained from two types of regions including: very smooth regions with lack of features and very complex uneven regions. Two qualifying tests, namely “Smoothness” and “Roughness”, are used to detect and remove the noisy motion vectors as follows:

2.2.1 Smoothness Test

The noisy motion vectors corresponding to the smooth regions such as sky image are detected by thresholding of the average of MAD (mean absolute difference) as:

$$MAD^{n}_{avg} < th_1, \quad (1)$$

where MAD^{n}_{avg} denotes the average of calculated MADs within the search area, during motion estimation of n^{th} block. th_1 is also defined as:

$$th_1 = MAD^{n}_{min} + T_1 \times \mathrm{Mean}(MAD^{n}_{avg}), \quad (2)$$

where MAD^{n}_{min} denotes the minimum value of computed MADs within the search area, during motion estimation of n^{th} block. T_1 is an experimentally obtained constant coefficient about 0.45 and $\mathrm{Mean}(MAD^{n}_{avg})$ denotes the average of MAD^{n}_{avg}, over all blocks of the frame.

2.2.2 Roughness Test

The noisy motion vectors corresponding to the rough regions are identified by another thresholding as:

$$MAD^{n}_{min} > th_2, \quad (3)$$

where threshold th_2 is defined adaptively as:

$$th_2 = T_2 \times \mathrm{Max}(MAD^{n}_{min}), \quad (4)$$

where T_2 is an experimentally constant coefficient about 0.45, and $\mathrm{Max}(MAD^{n}_{min})$ denotes the maximum value of MAD^{n}_{min}, over all blocks of the frame.

2.3 Global Motion Estimation

The global motion estimation unit produces a unique global motion vector (GMV) for each video frame, which represents the camera motion during the time interval of two frames. Since the MVs obtained from the image background tend to be very similar in both magnitude and direction, we used a clustering process to classify the motion field in two clusters corresponding to the background and foreground. The global motion induced by camera movement is determined by a clustering process that consists of the following steps.

- **Step 1)** Construct the histogram H of the valid local MVs. The value of H[x, y] is incremented by one each time the local MV(x, y) is encountered.
- **Step 2)** As long as the scene is not dominated by one single moving object, the cluster corresponding to the background blocks has the maximum votes in the clustering process. The max of this cluster is chosen as the global motion vector.

As an example, Figure3 shows the cluster located at (5, 12) receives the maximum vote, and the peak of this cluster yields the GMV.

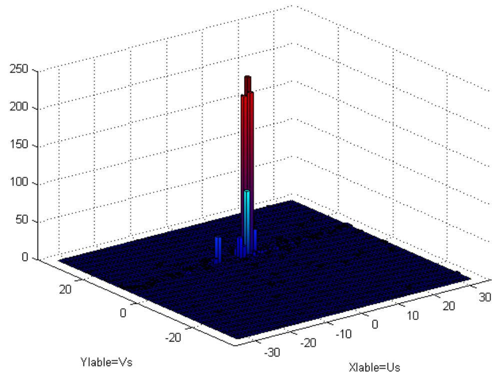

Figure 3- Clusters of motion field.

2.4 Unwanted Motion Estimation and Correction

An estimated GMV may consist of two major components: an intentional motion component (e.g. corresponding to camera panning) and unintentional motion component (e.g. corresponding to handshake). A good motion correction algorithm should only remove the unwanted motion while maintain the intentional motion. Assuming that the unwanted motion is corresponding to the high frequency components, the proposed algorithm uses a low pass filter to remove the unwanted motion component. A smooth motion vector (SMV) is resulted by filtering that resembles the intentional camera movement. The proposed method calculates the SMV in the form of first-order auto regression as

$$SMV(n) = \alpha\, SMV(n-1) + (1 - \alpha)\, GMV(n), \tag{5}$$

where $(0 \leq \alpha \leq 1)$ and the index n indicates the frame number. The reasons of using this first-order IIR filter are: (i) it can be used in real-time systems, (ii) it requires little memory, (iii) it involves little computations and (iv) the smoothed motions produced by the filter are satisfactory to human's eyes if a suitable value is selected for α. The parameter α can be regarded as the smoothing factor. A larger smoothing factor leads to a smoother, but a larger lag during intentional camera motion that makes artificially stabilized, image sequence. Therefore, a fixed value of α hardly leads to good stabilized image sequences. To avoid the lag of intentional movement and to smooth the unwanted camera motion efficiently, the following fuzzy adaptation mechanism of α is proposed.

2.4.1 Fuzzy Adaptation of Smoothing Filter

The smooth filtering is implemented on the vertical and horizontal components of the global motion vectors separately. The smoothing factor of filter is adjusted by a fuzzy system continuously. In facts, two fuzzy systems with a similar structure are used corresponding to the vertical and horizontal motion components. The fuzzy system has two inputs (Input1, Input2) and one output. The fuzzy inputs are defined as:

$$x_1 = \frac{1}{3}\sum_{i=n-2}^{n} \left|GMV_x(i)-GMV_x(i-1)\right|, \tag{6}$$

$$x_2 = \left|GMV_x(n)-GMV_x(n-3)\right|. \tag{7}$$

$$y_1 = \frac{1}{3}\sum_{i=n-2}^{n} \left|GMV_y(i)-GMV_y(i-1)\right|, \tag{8}$$

$$y_2 = \left|GMV_y(n)-GMV_y(n-3)\right|. \tag{9}$$

where x_1 and x_2 denote the inputs of fuzzy system used for adaptive filtering of the horizontal motion component and also y_1 and y_2 are the inputs of fuzzy system used for adaptive filtering of the vertical motion component. $GMV_x(n)$ and $GMV_y(n)$ indicate the horizontal and vertical components of the GMV of last frame. The fuzzy system inputs, Input1 (x_1, y_1) and Input2 (x_2, y_2), are used as quantitative representations of unwanted camera motion (noise) and intentional camera motion acceleration, respectively. The value of Input1 is proportional to the noise amplitude and the value of Input2 is proportional to the intentional motion acceleration. In the first-order IIR filter, a higher noise needs a larger smoothing factor for filtering. On the other hand, a large smoothing factor prevents tracking of intentional camera motion when acceleration. Therefore, the smoothing factor should be tuned carefully.

The proposed fuzzy system tunes the smoothing factor of the IIR filter adaptively according to the amount of noise and camera motion acceleration. In the proposed fuzzy system, trapezoidal and triangular membership functions (MFs) are used for each input and the output, respectively. The number of membership functions has been selected so as to obtain decent performance with as few membership functions as possible to maintain low system complexity. The experimentally designed input and output membership functions are shown in Figure 4. The constructed rule base is containing 30 rules as presented in Table I. The proposed fuzzy was implemented in MATLAB software while, the implication was set to *min* and the aggregation method to *max*. The defuzzification method was set to *centroid*. The output of fuzzy system defines the smoothing factor of IIR filter.

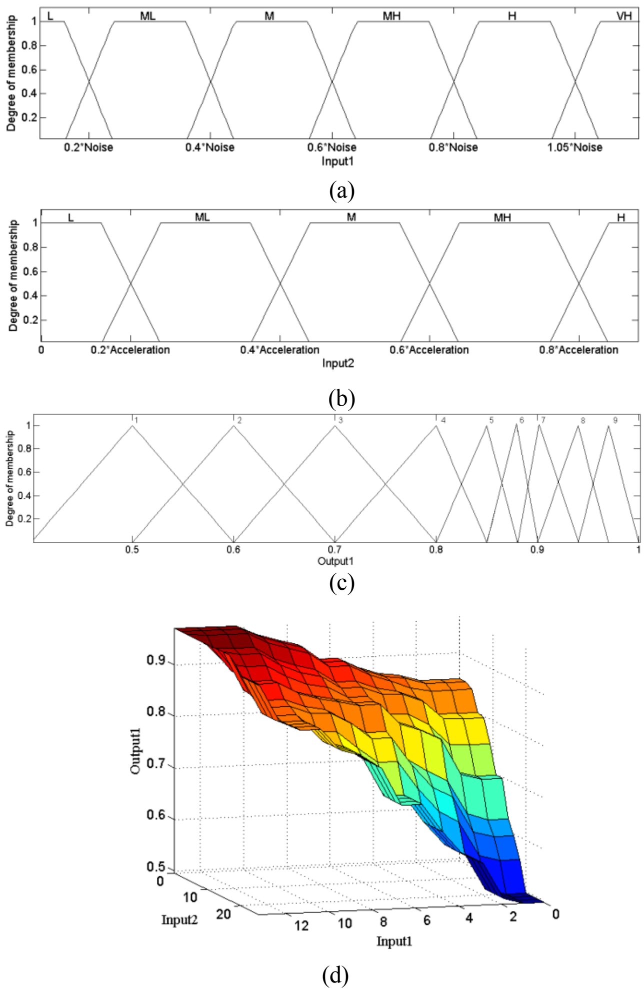

Figure 4 - The membership functions of fuzzy system.
(a) Input1, (b) Input2, (c) Output and (d) Surface.

2.4.2 Adaptive Fuzzy Membership Functions

Study on a number of video sequences has shown that the range of fuzzy inputs (Input1, Input2) is very variable on different video contents. Therefore, fixed MFs for the inputs of fuzzy system cannot provide a good performance for the stabilization system over all video contents. In order to have a good performance for the stabilization system over different video contents, it is proposed to adapt the membership functions of fuzzy inputs according to recently received video frames. The range of MFs for the fuzzy inputs i.e. [0, Input1(max)] and [0, Input2(max)] are modified adaptively as:

Input1 (max)= Max of input1 over recent 60 frames, $2 \le Input1 \le 12$ (10)

Input2(max)= Max of input2 over recent 60 frames, $2 \le Input2 \le 23$ (11)

If1.5(Input1(max))<Input2(max) Then Input1(max) =1.25(Input1(max))(12)

2.4.3 Motion Correction

After computing the smoothing factor α via the fuzzy system, SMV is calculated by (5). For the first three frames, a fixed Large value for α is used. After computing SMV, the unwanted motion vector (UMV) is obtained by

Table I: Rule Base for the Fuzzy System*.

	Input1					
Input2	L	ML	M	MH	H	VH
L	0.85	0.87	0.9	0.94	0.97	0.97
ML	0.8	0.85	0.87	0.9	0.94	0.97
M	0.7	0.8	0.85	0.87	0.9	0.97
MH	0.6	0.7	0.8	0.85	0.87	0.97
H	0.5	0.6	0.7	0.8	0.85	0.94

* L=Low, ML=Medium Low, M=Medium, MH=Medium High, H=High, VH=Very High

$$\mathrm{UMV}(n) = \mathrm{GMV}(n) - \mathrm{SMV}(n). \quad (13)$$

To restore the current frame to its stabilized position, we offset the current frame by the accumulated unwanted motion vector, AMV, defined by

$$\mathrm{AMV}(n)=\sum_{i=m}^{n}\mathrm{UMV}(i). \quad (14)$$

where m is the number of first frame since the last scene changes.

3. EXPERIMENTAL RESULTS

The performance of the proposed DVS is evaluated against 15 video sequences covering different types of scenes. Sample video sequences are available at [12], [13]. These sequences have a frame rate of 25 fps and a picture size of 352×288 pixels. Also the proposed fuzzy system has been tested with several synthetics data to simulate various situations. Some results for real and synthetics data are shown in figure 5. In this figure, 3 upper graphs are related to the real data and the rest graphs are related to the synthetics data. We worked with both gray-scale and color test sequences where in both cases motion estimation is done on the Y plane of YIQ color space. Stabilizer performance is assessed according to the smoothness of the resultant global motion compared to the original sequence and the gross movement preservation capability. Results are compared with provided results by presented algorithms in [9] and [11]. An adaptive IIR filtering technique and a fuzzy kalman system are proposed for motion correction in [9], [11], respectively. In each of our videos, there are jitters caused by car vibration, and shaky hands. Figure 5 presents the comparison between the original motions and the smoothed motions resulted by our DVS and presented algorithms in [9] and [11] for the real and synthetic data. Provided results by the presented algorithm in [9] show that changing the parameter α from 0.9 to 0.1 lead to undesirable movement in frame position, and also provided results by the presented algorithm in [11] show that we have close tracking of gross camera movements but at the cost of slightly reduced stabilization capabilities. Whereas results demonstrate that proposed fuzzy system provides expanded stabilization, while enables close tracking of gross camera movements. Our subjective experiments also demonstrated that human eyes have good visual perception to the stabilized video sequences by the proposed method due to removing unwanted camera motions (jitters) .

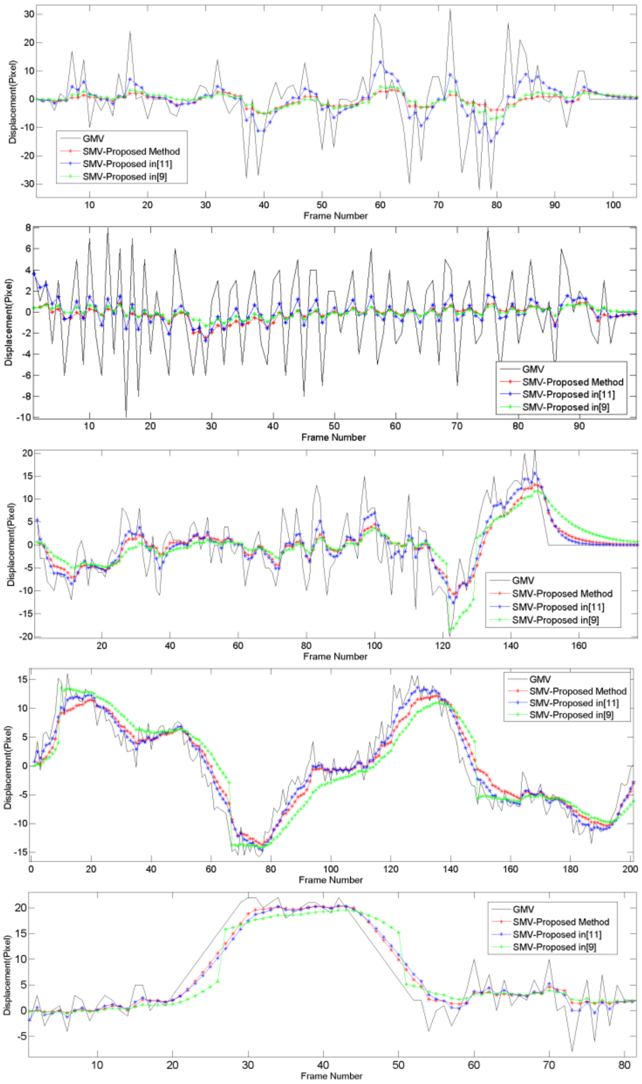

Figure 5- the absolute frame positions before and after stabilization for the real and synthetic data and their comparison with two previous presented algorithms.

4. CONCLUSION

In this paper, we proposed a computationally efficient digital video stabilization scheme using motion information obtained from a hybrid block based video encoder. Since some of the obtained motion vectors are not valid, an adaptive thresholding was developed to filter out valid motion vectors. To compute a global motion vector for each frame, the proposed stabilization technique effectively estimates the intentional camera motion by exploiting the characteristics of unwanted motions; an adaptive and low-complexity fuzzy IIR filter is proposed to fulfil two apparently conflicting requirements: close follow-up of the intentional camera movement and removal of the handshake. In order to improve stabilization performance, inputs membership functions of the fuzzy system are continuously adapted according to motion properties of a number of recently received video frames. Simulation results show a high performance for the proposed algorithm. With a low degree of computational complexity, the proposed scheme can be effectively used for the mobile video communications as well as the conventional video coding applications to improve the visual quality of digital video and to provide a higher compression performance.

REFERENCES

[1] A. Engelsberg and G. Schmidt, “A comparative review of digital image stabilizing algorithms for mobile video communications,” *IEEE Trans. Consumer Electron.,* vol. 45, no. 3, pp. 592-597, Aug. 1999.

[2] Y.-C. Peng, C.-K. Liang, H.-A. Chang, H. H. Chen, and C.-J Kao, “Integration of image stabilizer with video codec for digital video cameras,” in *Proc. Int. Symp. Circuits Syst*, Kobe, Japan, May, 2005, pp.4781–4784.

[3] W. Nan, H. Xiaowei, W. Gang and Y. Zhonghu, “ An approach of Electronic Stabilization Based on Binary image matching of Color Weight,” *2nd International Asia Conference on Informatics in Control, Automation and robotics (CAR)*, March 6-7. 2010, pp. 155-158.

[4] L. Marcenaro, G. vernazza, and C. S. Regazzoni, “Image stabilization algorithms for video surveillance applications,” *Proc. Int. Conf. on Image Processing*, vol. 1, pp. 349-352, Oct. 2001.

[5] A. A. Yeni and S. Ertürk, “Fast digital image stabilization using one bit transform based sub-image motion estimation,” *IEEE Transactions on Consumer Electronics*, vol. 51, no. 3, pp. 917– 921, 2005.

[6] S. Ertürk, "Multiplication-free one-bit transform for low complexity block-based motion estimation," *IEEE Sig. Proc. Letters*, vol. 14, no. 2, pp. 109-112, Feb. 2007.

[7] S.-J. Ko, S.-H. Lee, S.-W. Jeon, and E.-S. Kang, “Fast digital image stabilizer based on gray-coded bit-plane matching,” *IEEE Transactions on Consumer Electronics*, vol. 45, no. 3, pp. 598– 603, 1999.

[8] S. Ertürk, “Digital Image Stabilization with Sub-Image Phase Correlation Based Global Motion Estimation,” *IEEE Trans. Consumer Electron,* vol. 49, no. 4, pp. 1320-1325, Nov. 2003.

[9] S.-H. Yang and F.-M. Jheng, “An adaptive image stabilization technique,” In Proceedings of the *IEEE International Conference on Systems, Man, and Cybernetics* (SMC2006), Taipei, October 8 -11. 2006, pp. 1968-1973.

[10] M.K. Güllü and S. Ertürk, “Membership Function Adaptive Fuzzy Filter for Image Sequence Stabilization,” *IEEE Trans. Consumer Electron,* vol. 50, no. 1, pp. 1-7, Feb. 2004.

[11] N. Kyriakoulis and A. Gasteratos, “A Recursive Fuzzy System for Efficient Digital Image Stabilization,” *Hindawi Advances in Fuzzy Systems*,Vol. 8, Jan. 2008.

[12] Shaky Car, matlabroot\toolbox\vipblks\vipdemos\shaky_car.avi

[13] http://www.jnack.com/adobe/photoshop/videostabilization/