

# Chapter 5

## Fuzzy-Based Digital Video Stabilization in Static Scenes

Margarita Favorskaya and Vladimir Buryachenko

**Abstract** In recent years, a digital video stabilization improving the results of hand-held shooting or shooting from mobile platforms is the most popular approach. In this chapter, the task of digital video stabilization in static scenes is investigated. The unwanted motion caused by camera jitters or vibrations ought to be separated from the objects motion in a scene. Our contribution connects with the development of deblurring method to find and improve the blurred frames, which have strong negative influence on the following processing results. The use of fuzzy Takagi-Sugeno-Kang model for detection the best local and global motion vectors is the novelty of our approach. The quality of test videos stabilization was estimated by Peak Signal to Noise Ratio (PSNR) and Interframe Transformation Fidelity (ITF) metrics. Experimental data confirmed that the ITF average estimations increase up on 3–4 dB or 15–20 % relative to the original video sequences.

**Keywords** Digital video stabilization · Deblurring · Motion estimation · Fuzzy logic · Scene alignment

### 5.1 Introduction

The unintentional video camera motion during hand-held shooting or shooting by cameras, which are maintained on the unstable platforms, decreases a quality of video sequence that has a negative effect on the following frames processing.

---

M. Favorskaya (✉) · V. Buryachenko  
Department of Informatics and Computer Techniques, Siberian State Aerospace University,  
31 Krasnoyarsky Rabochy Avenue, Krasnoyarsk 660014, Russian Federation  
e-mail: favorskaya@sibsau.ru

V. Buryachenko  
e-mail: buryachenko@sibsau.ru

The use of stabilized techniques is widely spread in video surveillance [1] and video encoding [2]. Methods of videos stabilization are classified as mechanical, optical, electronic, and digital. The mechanical stabilization systems included gyroscopes, accelerometers, etc., and were used on the early stage of video cameras development [3]. The optical stabilization systems use prisms or lens of moving assembly for tuning of light length way through camera lens systems. Such technical realization is not suitable for small sizes mobile cameras. The electronic stabilization systems detect the camera jitters through their sensors, when the light hits in Charge-Coupled Device (CCD). It has the advantage against the optical stabilization by reducing of lens complexity and price.

The Digital Video Stabilization (DVS) approach is achieved by the synthesis of a new imagery based on removal of unintentional motions between key frames and the reconstruction of frame boundaries after frame stabilization. The DVS based on the algorithmic improvement became the most appropriate decision in modern compact video devices. However, the DVS algorithms ought to be robust to different scene contents, moving objects, and luminance changing. The complexity of this task connects with separation of the objects motion from camera jitters in static scenes.

Our contribution connects with the improvement of some blurred frames in original video sequence, the development of DVS method based on the Takagi-Sugeno-Kang (TSK) model for improvement of motion vectors clustering, and the application of scene alignment procedure into static scenes.

The chapter has been structured as follows. In Sect. 5.2, a brief description of the existing approaches for digital video stabilization previously in static scenes is provided. Method for automatic blurred image detection and compensation is discussed in Sect. 5.3. Section 5.4 describes the proposed method of videos stabilization based on Block-Matching Algorithm (BMA) using the fuzzy logic approach in the selected image area. Experimental numerical results of the proposed approach are presented in Sect. 5.5. Conclusions and future research are drawn in Sect. 5.6.

## 5.2 Related Work

Video stabilization includes three main stages: motion estimation, motion smoothness, and frame correction. Approaches of motion estimation are directed on reduction of computational cost using fast BMA [4], limited pre-determined regions [5], or feature tracking [6]. Tanakian et al. [4] proposed the integrated system of video stabilizer and video encoder based on the BMA for the Local Motion Vectors (LMVs) detection, histogram analysis for Global Motion Vector (GMV) building, and Smooth Motion Vector (SMV) calculation for intentional motion correction. Tanakian et al. suggested a low pass filtering to remove a high frequency component of the intentional motion. Equation 5.1 approximates the SMVs using a first-order auto regression function, where  $\alpha$  is a smoothing factor,  $0 \leq \alpha \leq 1$ ;  $n$  is a frame number.

$$|SMV_n| = \alpha |SMV_{n-1}| + (1 - \alpha) |GMV_n| \quad (5.1)$$

Also Tanakian et al. proposed the rule to chose  $\alpha$  value ( $\alpha = 0.1$  or  $\alpha = 0.95$ ) in dependence of the SMVs and the GMVs magnitudes in previous frames. After following researches, a fuzzy system for tuning of smoothing factor  $\alpha$  was suggested according to noise and the possible camera motion acceleration [5]. Triangular and trapezoidal membership functions were used for adaptive filtering of horizontal and vertical motion components between  $(n - 3)$ ,  $(n - 2)$ ,  $(n - 1)$ , and  $n$  frames.

Another approach was applied by Acharjee and Chaudhuri in [7] as a Three Step Search BMA, which provides the fuzzy membership values. The fuzzy membership value is calculated for each macro block according to its intensity characteristics (lighter or darker) previously in edges of an image. Such restrictions reduce the computational cost of full Three Step Search BMA, while the PSNR values are almost saved. In research [8], a fuzzy logic model to evaluate a quality of matching between a pair of Scale-Invariant Feature Transform (SIFT) descriptors was proposed. A fuzzy logic model used two error measures (Euclidean distance between expected and real points and angle between two local motion vectors) as inputs and the single quality index in the range  $[0, 1]$  as output. A final decision of a quality of points' matching was estimated using a Sugeno model [9].

A fuzzy Kalman compensation of the GMV in the log-polar plane was proposed in research [10]. Due to special features of the log-polar plane, the GMV was calculated as the average value of the four LMVs. Then the GMV displacements were imported into the fuzzy Kalman system. The fuzzy system was tested with several types of Membership Functions (MFs) and different aggregation and defuzzification methods. Some original approaches may be found in the researches dedicating to video stabilization by use a principal component analysis [11], an independent component analysis [12], a probabilistic global motion estimation based on Laplacian two-bit plane matching [13], wavelet transformations [14], the calculation of statistical functions, mean and variance of pixels in each block of the BMA [15], etc.

The blurred frames have negative influence on stabilization of video sequence. The reasonable approach connects with removal such blurred frames before the application of stabilization algorithms [16]. The different deblurring methods are mentioned below:

- Deblurring methods for a single image reconstruction are applied usually the uniform deblurring core [17, 18]. Such methods cannot be applied directly for frames from video sequence in spatio-temporal domain. In recent years, 3D deblurring core was proposed to describe a spatio-changing core for a frame [19]. Gupta et al. [20] proposed to present the camera motion using motion density functions. Such methods are not reliable for real noisy and/or compressed video sequences with multiple moving objects in a scene.
- Deblurring methods for several images reconstruction are based on different approaches such as the numerical method for calculation of a blurring model [21], the use of the point spread function to restore the original images with

minimal energy [22], or the interpolation method in order to increase the sharpness all frames including the deblurred frames [23].

- Methods based on selection of “suitable” images guess the overlapping of several good input images in order to receive the best single image. However, the superposition of real frames is not a trivial task because of blurring and luminance differences of selected frames.

Consider the proposed deblurring method as a crucial issue of pre-processing in the DVS task.

### 5.3 Method of Frame Deblurring

The proposed method applies the anisotropic Gauss filter with adaptive automatic selection of region sizes and includes the following steps:

1. The automatic estimation of blurred frames based on gradient information.
2. The detection of textured and smoothness regions in a frame.
3. The analysis of edge information by use a Sobel filter.
4. The application of anisotropic Gauss filter with automatic mask selection in textured regions.
5. The application of unsharp mask in smoothness region.
6. The synthesis of result frame.

Let us notice that a blurring happens not always, but during high jitters, fast motion objects, or continuous camera exposition. Therefore, not all frames are blurred, and it is required to detect, which frames are blurred into the analyzed part of video sequence.

Introduce a measure of sharp estimation based on gradient information into full frame and pre-determined blurring threshold value  $T$ . A blurring degree in current frame is estimated by Eq. 5.2, where  $T$  is a blurring threshold value,  $0 \leq T \leq 1$ ,  $g_n$  is a gradient function of frame,  $I_{ij}$  is value of intensity function of frame in a pixel with coordinates  $(i, j)$ ,  $N$  and  $M$  are the frame sizes,  $K$  is a number of previous key frame.

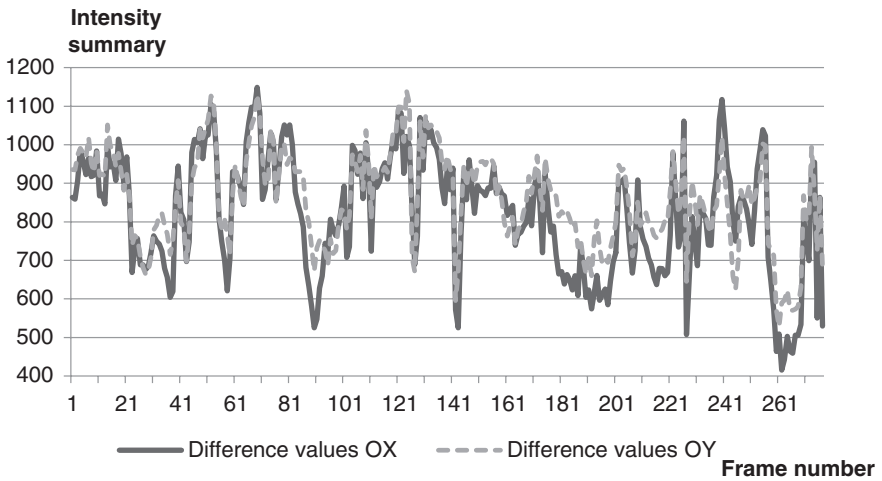
$$\sum_{i=1}^M \sum_{j=1}^N g_n^2(I_{ij}) < T$$

$$\times \max \left\{ \sum_{i=1}^M \sum_{j=1}^N g_{n-K}^2(I_{ij}), \sum_{i=1}^M \sum_{j=1}^N g_{n-(K+1)}^2(I_{ij}), \dots, \sum_{i=1}^M \sum_{j=1}^N g_{n-1}^2(I_{ij}) \right\}$$
(5.2)

Examples of detected frames with high and low blurring degree are situated in Fig. 5.1.



**Fig. 5.1** Video sequence “Sam\_1.avi”: **a** frame 59 with high sharp, **b** frame 68 with low sharp, **c** the increased fragment of frame 59, **d** the increased fragment of frame 68



**Fig. 5.2** A blurring estimation of video sequence “Sam\_1.avi”

For video sequence “Sam\_1.avi”, the plots of blurring degree for frames 1–265 are situated in Fig. 5.2. The maximum values mean the availability of blurred frames. This scene contains fast motion and significant jutting.

For detection of textured and smoothness regions in a frame, the frame differences are estimated in a slicing window  $5 \times 5$  pixels by Eq. 5.3, where  $\beta_L(x, y)$  is a blurring degree into pixel with coordinates  $(x, y)$ .

$$\beta_L(x, y) = \frac{1}{25} \sum_{i=1}^5 \sum_{j=1}^5 \left( I_{i,j}^2 - I_{(i+1),(j+1)}^2 \right) \quad (5.3)$$

According to the received values  $\beta_L(x, y)$ , the binary map  $B_m(x, y)$  is calculated in order to estimate the textured and smoothness regions into a frame by Eq. 5.4, where  $T_{fl}$  is an automatic chosen threshold value in dependence of total  $g_n$  value provided by Eq. 5.2.

$$B_m(x, y) = \begin{cases} 0 & \text{if } \beta_L(x, y) < T_{fl} \\ 1 & \text{in other cases} \end{cases} \quad (5.4)$$

The analysis of edge information can be realized by Sobel filter or other filters for edge estimations. If a density of edges is high into a unit region, then this region is considered as a textured region. In such case, the anisotropic Gauss filter is applied. The core of this filter is adaptively tuned for pixel closed to an edge, pixel near the edge, and pixel far from an edge. This procedure is required in order to exclude the sharp edge pixels from such processing. The parameters of anisotropic Gauss filter are calculated with different scaling along axes OX and OY by Eq. 5.5, where  $\sigma_x$  and  $\sigma_y$  are standard deviations chosen in dependence of remote pixel  $(x, y)$  from an edge.

$$g(x, y; \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left\{ -\frac{1}{2} \left( \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right\} \quad (5.5)$$

For smoothness regions, the unsharp mask, which is based on the subtraction the blurred frame from the original frame, is used. The sizes of unsharp mask are changed dynamically in dependence on analyzed region sizes in a frame. A synthesis of the deblurred frame is realized by considering the edge, improved textured, and improved smoothness information.

A method of frame deblurring was developed under the assumption that a number of blurred frames into the analyzed part of video sequence is not large, one or two from 25–30 frames. Therefore, a computational cost of this procedure is not high. However, the influence on the final stabilization result is positive.

## 5.4 Fuzzy-Based Video Stabilization Method

The proposed video stabilization method involves three main stages: the LMVs estimation, the GMVs smoothness, and the frames correction. The LMVs detection with following improvement by the TSK model is discussed in Sect. 5.4.1. Section 5.4.2 provides the smoothness GMVs building. In Sect. 5.4.3, the static scene alignment is considered.

### 5.4.1 Estimation of Local Motion Vectors

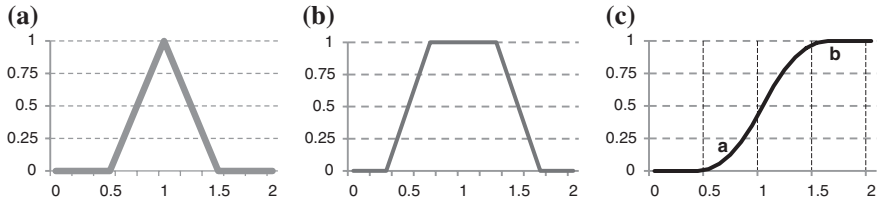
For the LMVs estimations, many approaches with various computational costs can be applied. The experiments show that the BMA provides fast motion estimations with appropriate accuracy in static scene. First, the current frame is divided in the non-crossed blocks with similar sizes (usually  $16 \times 16$  pixels), which are defined by the intensity function  $I_t(x, y)$ , where  $(x, y)$  are coordinates,  $t$  is a discrete time moment. Second, for each block in small neighborhoods  $-S_x < d_x < +S_x$  and  $-S_y < d_y < +S_y$ , the most similar block into the following frame  $I_{t+1}(x + d_x, y + d_y)$  is searched. The similarity is determined by a minimization of the error functional  $e$  according to the used metric. Usually three metrics are applied such as a Sum of Absolute Differences (SAD), a Sum of Squared Differences (SSD), and a Mean of Squared Differences (MSD) (Eq. 5.6), where  $d_x$  and  $d_y$  are the block displacement in directions OX and OY, respectively,  $n$  is a number of analyzed surrounding blocks.

$$\begin{aligned}
 e_{SAD}(d_x, d_y) &= \sum_{x=1}^N \sum_{y=1}^N |I_{t+1}(x, y) - I_t(x + d_x, y + d_y)| \\
 e_{SSD}(d_x, d_y) &= \sum_{x=1}^N \sum_{y=1}^N (I_{t+1}(x, y) - I_t(x + d_x, y + d_y))^2 \\
 e_{MSD}(d_x, d_y) &= \frac{1}{n \times n} \sum_{x=1}^N \sum_{y=1}^N (I_{t+1}(x, y) - I_t(x + d_x, y + d_y))^2
 \end{aligned} \tag{5.6}$$

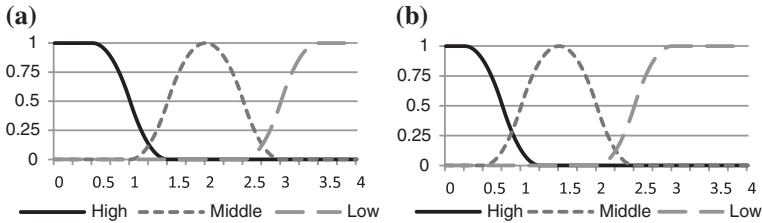
Vector  $\mathbf{V}(d_x, d_y)$ , for which the error functional  $e$  ( $e_{SAD}(d_x, d_y)$ ,  $e_{SSD}(d_x, d_y)$ , or  $e_{MSD}(d_x, d_y)$ ) has the minimum value, is considered as the displacement vector for the selected block. The basic BMA referred as Full Search (FS) has a disadvantage of high computer cost. Some modifications of the FS exist such as Three-Step Search (TSS), Four-Step Search (FSS), Conjugate Direction Search (CDS), Dynamic Window Search (DSW), Cross-Search Algorithm (CSA), Two-Dimensional Logarithmic Search (TDLS), etc.

After the LMVs building, it is needed to determine, which LMVs describe an unwanted camera motion and which LMVs concern to objects' motion in a scene. The proposed model is based on triangular, trapezoidal, and  $S$ -shape memberships in terms of fuzzy logic to partitioning the LMVs. The views of memberships are presented in Fig. 5.3, where parameters  $a$  and  $b$  of  $S$ -shape membership are fitted empirically. The recommendations based on the experimental results are the following: to use  $a = 0.5$  and  $b = 1.5$  for non-noisy videos and  $a = 0.75$  and  $b = 1.75$  for noisy videos. The inputs of fuzzy logic model have two error measures:

- The Euclidean distance between the expected and the real point estimated in one of the SAD, the SSD, or the MSD metrics (magnitude of vectors)  $\mathbf{E}' = (e_1, e_2, \dots, e_i, \dots, e_n)$ .



**Fig. 5.3** View of memberships in fuzzy logic model: **a** triangular, **b** trapezoidal, **c** S-shape



**Fig. 5.4** View of S-shape memberships: **a** for non-noisy video sequence, **b** for noisy video sequence

- The angle between two local motion vectors  $C' = (c_1, c_2, \dots, c_i, \dots, c_n)$ , where  $i = 1 \dots n$ .

Equation 5.7 provides the error deviations  $d_i^e$  and  $d_i^c$  similar to research [6], where  $M_E$  and  $M_C$  are the median values of  $E'$  and  $C'$  sets, respectively.

$$d_i^e = e_i / M_E \quad d_i^c = c_i / M_C \tag{5.7}$$

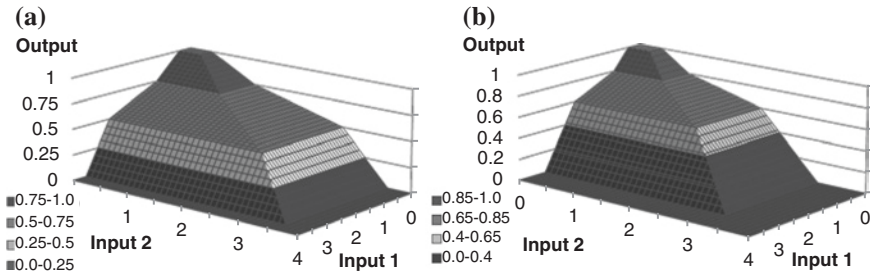
Values of error deviations  $d_i^e$  and  $d_i^c$  are mapped into three different classes of accuracy: high, medium, and low. The lower values of error deviations are mapped to the best classes, and otherwise. If membership functions are overlapped, then more good definition from the input fuzzy sets is chosen.

The output of fuzzy logic model indicates a final reliability of matching quality using the TSK model [9]. Such zero-order fuzzy model infers the quality index (a value in the range [0, 1]). The quality of the points' matching is classified into four categories: excellent, good, medium, and bad. Each of these four classes is mapped into a set of constant values 1.0, 0.75, 0.5, 0.0, respectively. Views of triangular, trapezoidal, and S-shape memberships are situated in Fig. 5.3.

The recommended S-shape functions are situated in Fig. 5.4.

The output of fuzzy logic model indicates a final reliability of estimations for a quality of the matching using the TSK model. The quality index is a value in the range [0, 1]. It shows a quality of LMVs, which are clustered into four classes: excellent, good, medium, and bad. The “IF-THEN” fuzzy rules defined for two inputs (error deviations  $d_i^e$  and  $d_i^c$ ) are the following:





**Fig. 5.5** View of TSK models: **a** for non-noisy video sequence, **b** for noisy video sequence

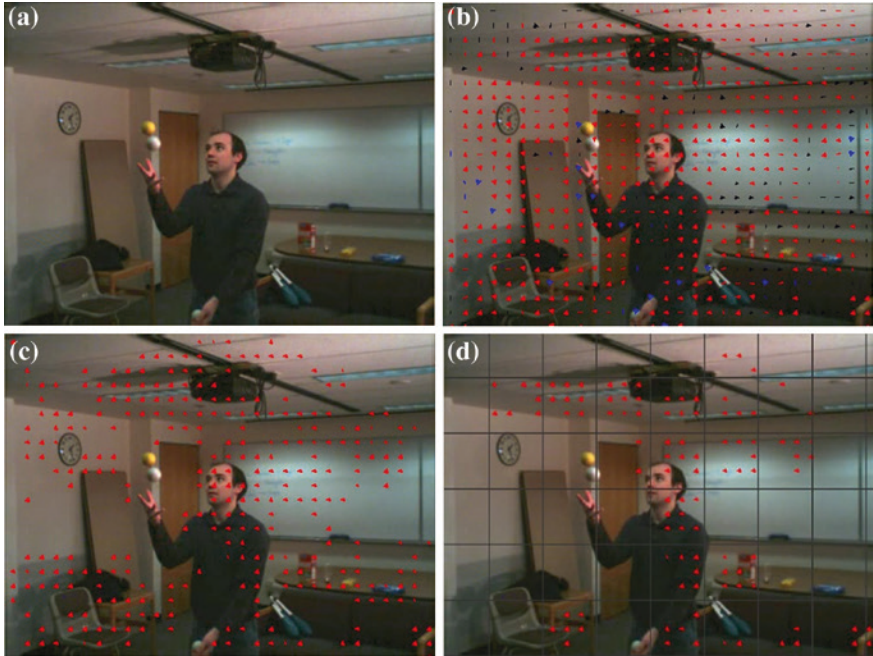
- IF (both inputs = “high”) THEN (quality = “excellent”).
- IF (one input = “high” AND other input = “medium”) THEN (quality = “good”).
- IF (both inputs = “medium”) THEN (quality = “medium”).
- IF (at least one input = “low”) THEN (quality = “bad”).

Each of these four classes is mapped into a set of the constant values (1.0, 0.75, 0.5, 0.0) [9]. During our experiments, the results for noisy video sequences were received with a set of the constant values (1.0, 0.85, 0.65, 0.0). The TSK models for non-noisy and noisy video sequences with the sets of constant values (1.0, 0.75, 0.5, 0.0) and (1.0, 0.85, 0.65, 0.0) are show in Fig. 5.5 [24]. The TSK model permits to discriminate the LMVs with excellent and good quality and detect the best LMVs (with excellent and good values of indexes) in order to improve a final result.

Our following researches permitted to speed the LMVs calculation for both types of video sequences in the static scenes. Introduce an initial procedure, which will put an invisible grid on each frame adaptively to the frame sizes with 30–50 cells.

The sizes of such grid are less in order to reject the boundary areas of frame, which are more stressed to artifacts of instability. For five first frames in a scene, the LMVs estimations and their improvements by the TSK model are calculated for all cells of a grid. For each cell, the information of reliable LMVs is accumulated under the condition, that 4–16 reliable LMVs are determined into a cell. According to a scene background, such several cells can be selected for following analysis. Therefore, the LMVs of unwanted motion are calculated only in the selected cells that permits to avoid the challenges of luminance changing or moving foreground objects and reduce the number of analyzing cells in 1.5–3 times. Figure 5.6 provides such adaptive and fast technique for frame number 140 from video sequence “If\_juggle.avi”.

The TSK model discriminates the excellent and good results well. The selection of the best points with excellent and good values of indexes improves the final results.



**Fig. 5.6** The adaptive technique for LMVs estimation in a static scene, video sequence 'lf\_juggle.avi': **a** the original frame 140; **b** all calculated LMVs; **c** the reliable LMVs in the whole frame based on TSK model; **d** the reliable LMVs in the selected cells of imposed grid

### 5.4.2 Smoothness of GMVs Building

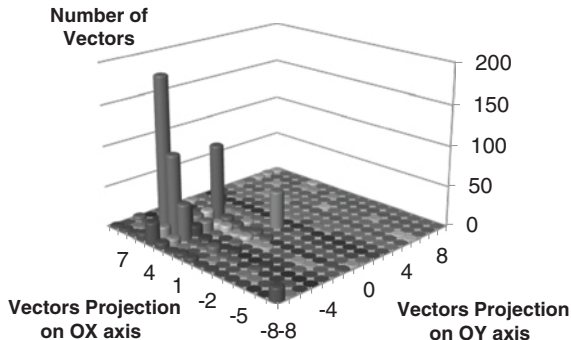
The global motion caused by camera movement is estimated for each frame by use a clustering model. The LMVs of background are very similar on magnitudes and directions but essentially different from objects' motion in foreground. The following procedure classifies the motion field into two clusters: the background and the foreground motions:

- Step 1. The histogram  $H$  is built, which includes only valid LMVs.
- Step 2. The LMVs are clustered by a similar magnitudes criterion.
- Step 3. The LMV with a maximum magnitude from background motion cluster is chosen as GMV.

The example of a histogram with valid LMVs is presented in Fig. 5.7.

Any GMV includes two major components: the real motion (for example, a panning) and the unwanted motion. Usually the unwanted motion corresponds to high frequency component. Therefore, a low-frequency filtering can remove the unwanted motion. The model proposed in research [4] forms the SMV calculating by Eq. 5.1. The low-pass filter of the first order needs in low computational

**Fig. 5.7** Example of a histogram with valid LMVs



resources and may be used in a real-time application. To improve these results, a similar fuzzy logic model from Sect. 5.4.1 was used for clustering the intentional and the non-intentional GMVs. The adapting tuning procedure of a smoothing factor  $\alpha$  based on analysis of previous 25 frames was proposed. First, Eq. 5.8 calculates a Global Difference  $GDiff_k$ , where  $|GMV_i|$  is the magnitude of global motion vector in a frame  $i$ ,  $k > 25$ .

$$GDiff_k = \sum_{i=k-25}^k ||GMV_i| - |GMV_{i-1}|| \quad (5.8)$$

Second,  $\alpha$  value is chosen by Eq. 5.9, where  $\alpha_{max} = 0.95$  and  $\alpha_{min} = 0.5$  are maximum and minimum empirical values.

$$\alpha_k = \begin{cases} \frac{GDiff_k}{|GMV_{max}|} \times \frac{\alpha_{max}}{\alpha_{min}} & \text{if } GDiff_k < |GMV_{max}| \\ \alpha_{max} & \text{if } GDiff_k \geq |GMV_{max}| \end{cases} \quad (5.9)$$

In any case, the result from Eq. 5.9 is rounded to  $\alpha_{max}$ .

### 5.4.3 Static Scene Alignment

For each frame after the smooth factor  $\alpha$  calculation, the module of smooth motion vector  $SMV_n$  using Eq. 5.1 is determined. The module of an Undesirable Motion Vector (UMV)  $UMV_n$  is calculated by Eq. 5.10.

$$|UMV_n| = |GMV_n| - |SMV_n| \quad (5.10)$$

In the development of a scene alignment, the direction of  $SMV_n$  is normalized up to 8 directions with interval of  $45^\circ$ . For restoration of current frame, pixels are shifted on a value of an Accumulated Motion Vector (AMV)  $AMV_n$  of unwanted motion by Eq. 5.11, where  $m$  is the number of a current key frame in video sequence.

$$AMV_n = \sum_{i=m}^n |UMV_i| \quad (5.11)$$

The stabilized location of frame is determined from previous frames beginning from the current key frame.

## 5.5 Experimental Results

Six video sequences received by the static camera shooting were used during experiments. The titles, URL, and snapshots of these investigated video sequences are presented in Table 5.1. All experiments were executed by the own designed software tool “DVS Analyzer”, v. 2.04. The software tool “DVS Analyzer” has two modes: the pseudo real-time stabilization of video sequences, which are broadcasted from the surveillance cameras (the simplified processing), and the unreal-time stabilization of available video sequences (the intelligent processing).

The architecture of the software tool includes the extended set of program modules, which can be developed independently each from others. The Pre-processing Module, the Motion Estimation Module, the Motion Compensation Module, the Motion Inpainting Module, the Module of Quality Estimation, the Core Module, and the Interface Module are the main components of the software tool “DVS Analyzer”. The software tool “DVS Analyzer”, v. 2.04 was designed in the Rapid Application Development Embracadero RAD Studio 2010. Some external software tools were used such as the libraries “Video For Windows” for initial processing and “AlphaControls 2010 v7.3” for enhanced user interface, and a video codec “K-Lite Codec Pack”, v. 8.0.

The experimental graphics for motion estimation and compensation of video sequences situated in Table 5.1 are represented in Fig. 5.8. The experimental plots for stabilization quality of these video sequences are represented in Fig. 5.9.

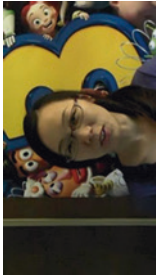
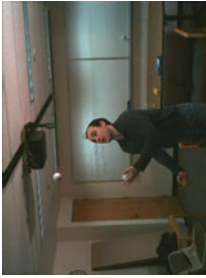

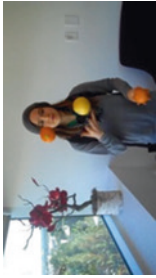
As it is shown from Fig. 5.9, the PSNR estimations of the stabilized video sequences are always higher than the PSNR estimations of the original video sequences.

The objective estimation of video stabilization quality was calculated by the PSNR metric between current frame  $I^{cur}$  and key frame  $I^{key}$  expressed in Eqs. 5.12–5.13, where  $MSE$  is a mean-square interframe error,  $I_{max}$  is a maximum of pixel intensity,  $m$  and  $n$  are sizes of frame.

$$MSE = \frac{1}{m \times n} \sum_{y=1}^m \sum_{x=1}^n \left( I^{cur}(x, y) - I^{key}(x, y) \right)^2 \quad (5.12)$$



$$PSNR = 10 \log_{10} \left( \frac{I_{max}^2}{MSE} \right) \quad (5.13)$$

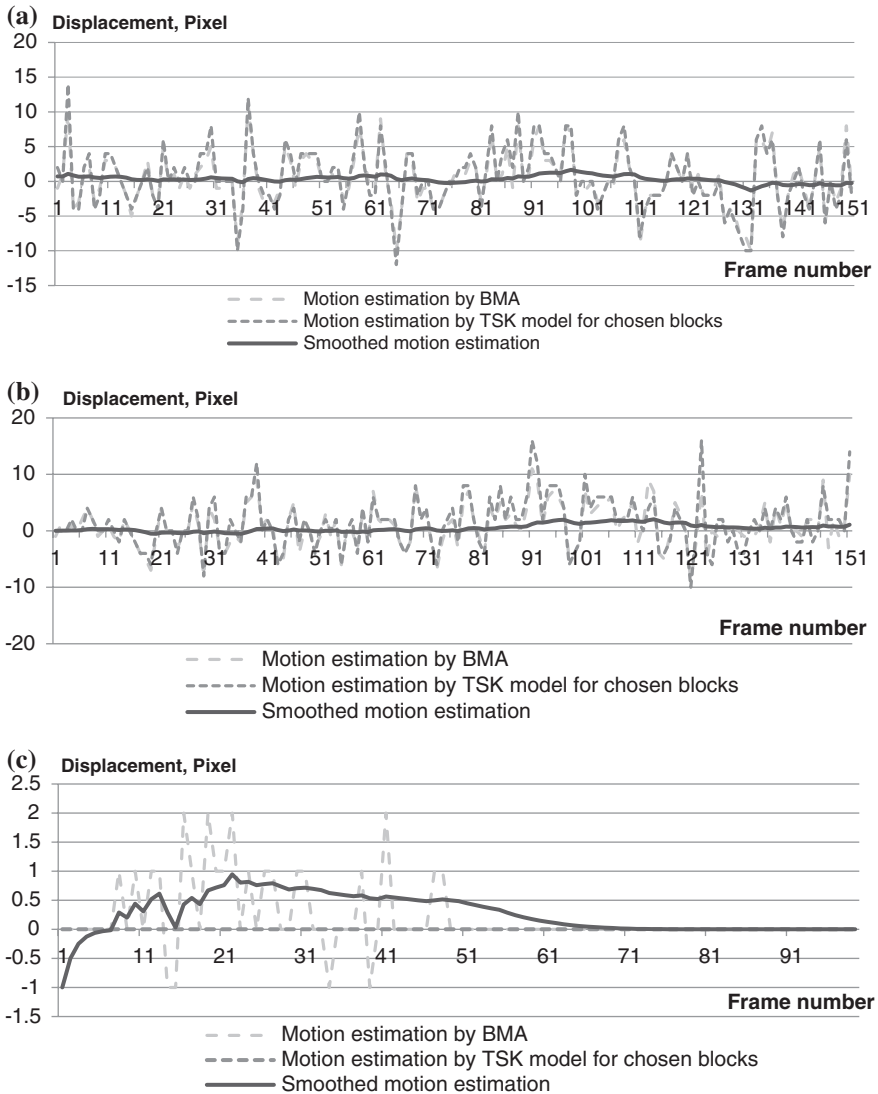
**Table 5.1** Description of investigated static scenes

Title, URL	Snapshot	Resolution	Frames	Motion type
"SANY0025_xvid.avi" <a href="http://cpl.cc.gatech.edu/projects/projects/videostabilization/">http://cpl.cc.gatech.edu/projects/videostabilization/</a>		640 × 360	445	Slow motion of camera, large object motion
"if_juggle.avi" <a href="http://cpl.cc.gatech.edu/projects/videostabilization/">http://cpl.cc.gatech.edu/projects/videostabilization/</a>		480 × 360	460	Static camera, fast motion of small objects
"akiyo.avi" <a href="http://see.xidian.edu.cn/vips/database_Video.html">http://see.xidian.edu.cn/vips/database_Video.html</a>		352 × 288	300	Static camera, large object motion
"EllenPage_Juggling.avi" <a href="http://www.youtube.com/watch?v=8YNUSCX_akk">http://www.youtube.com/watch?v=8YNUSCX_akk</a>		1280 × 720	430	Static camera, fast motion of small objects

(continued)

Table 5.1 (continued)

Title, URL	Snapshot	Resolution	Frames	Motion type
“Butovo_synthetic.avi” <a href="http://youtu.be/0oeIZ04sXu0">http://youtu.be/0oeIZ04sXu0</a>		640 × 480	748	Static camera, strong vertical and horizontal camera jittings
“road_cars_krasnoyarsk.avi” <a href="http://www.youtube.com/watch?v=pJ84Pwpbl_Y">http://www.youtube.com/watch?v=pJ84Pwpbl_Y</a>		852 × 480	430	Static camera, several moving objects, high deep of scene



**Fig. 5.8** Plots of motion estimation and compensation results in static scenes: **a** “SANY0025\_xvid.avi”, **b** “If\_juggle.avi”, **c** “akiyo.avi”, **d** “EllenPage\_Juggling.avi”, **e** “Butovo\_synthetic.avi”, **f** “road\_cars\_krasnoyarsk.avi”

The PSNR metric is useful for estimations between adjacent frames. A quality of the ITF metric provides the objective estimation in whole video sequence. The ITF of stabilized video sequence is higher than the ITF of original video sequence. This parameter is calculated by Eq. 5.14, where  $N_{fr}$  is a frame number in video sequence.

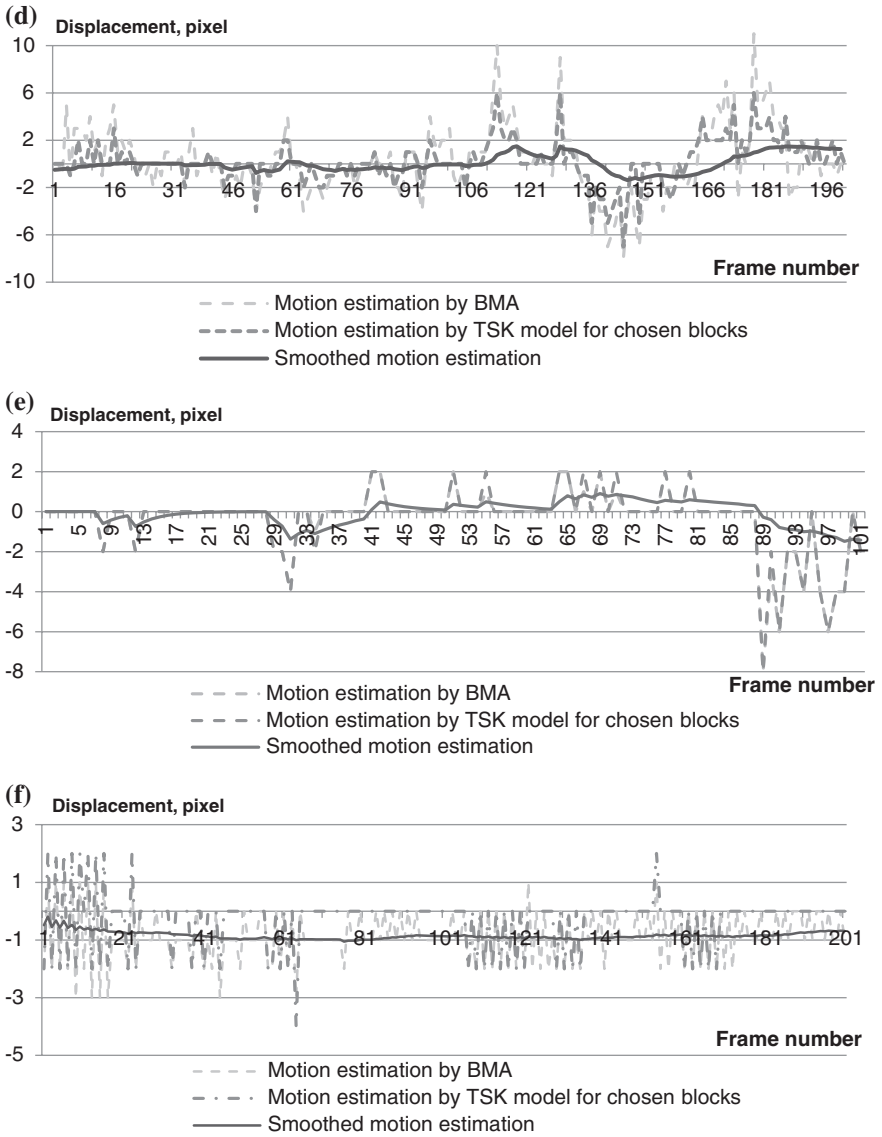
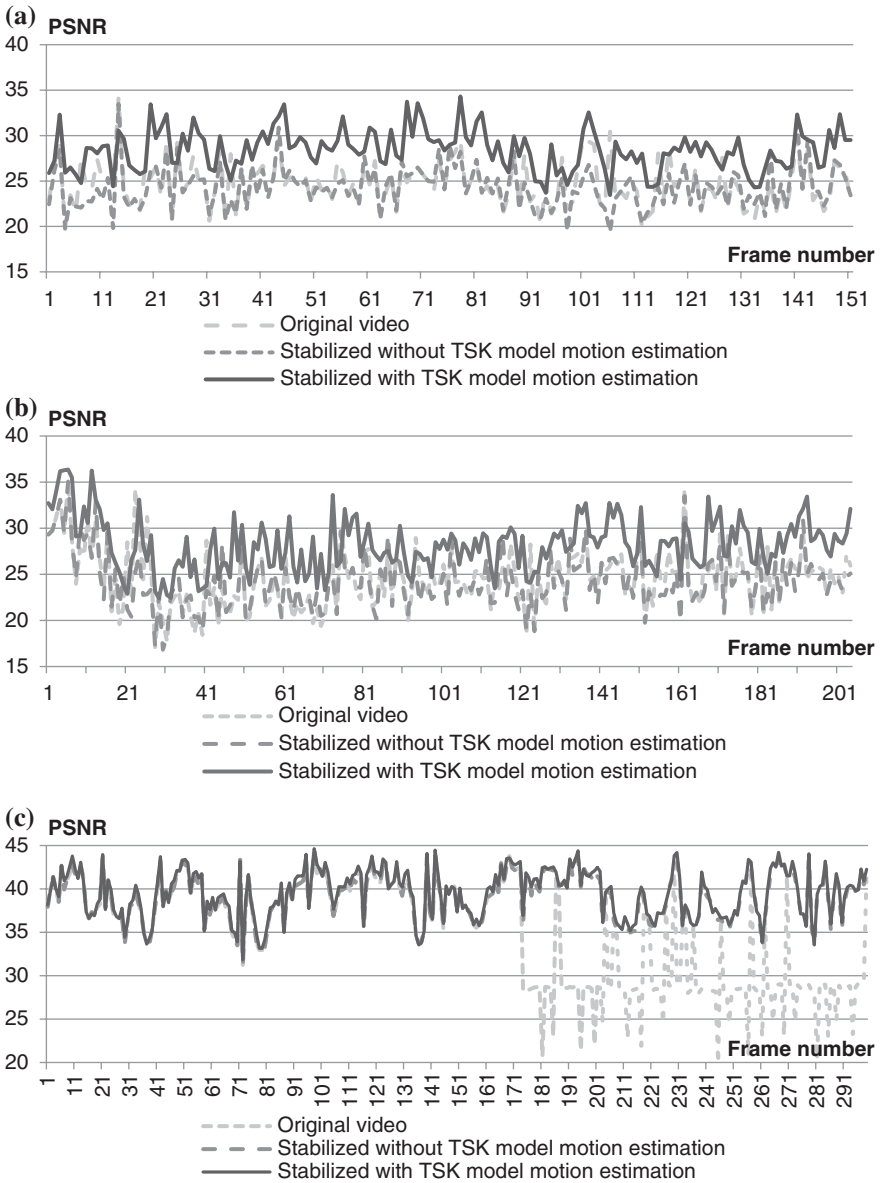


Fig. 5.8 (continued)

$$ITF = \frac{1}{N_{fr}} \sum_{k=0}^{N_{fr}} PSNR_k \tag{5.14}$$





**Fig. 5.9** Plots of stabilization quality in static scenes: **a** “SANY0025\_xvid.avi”, **b** “lf\_juggle.avi”, **c** “akiyo.avi”, **d** “EllenPage\_Juggling.avi”, **e** “Butovo\_synthetic.avi”, **f** “road\_cars\_krasnoyarsk.avi”

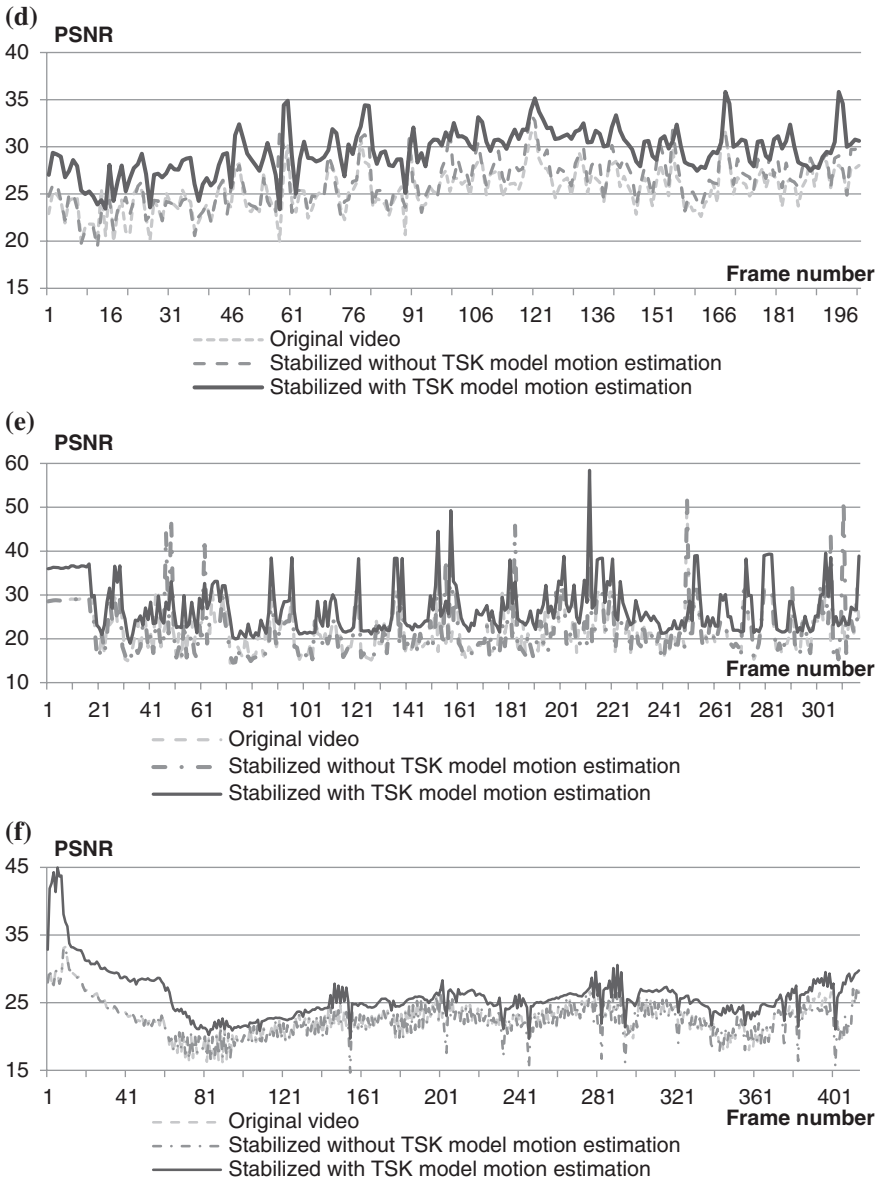


Fig. 5.9 (continued)

Table 5.2 contains the ITF estimations for video sequences: original, without and with TSK model application.

As it seems from Table 5.2, the video stabilization results are different for various video sequences because of varied foreground and background content,

**Table 5.2** ITF estimations for static scenes

Video sequence	ITF estimations, dB		
	Original	Without TSK model	With TSK model
“SANY0025_xvid.avi”	20.5389	21.09076	23.79189
“lf_juggle.avi”	24.30286	24.37177	28.06012
“akiyo.avi”	35.92952	39.14661	39.53257
“EllenPage_Juggling.avi”	24.65855	25.23049	28.58255
“Butovo_synthetic.avi”	22.26415	27.19789	27.20789
“road_cars_krasnoyarsk.avi”	22.70482	22.80707	25.91258

**Table 5.3** Comparison of stabilization algorithms for static scenes

Video sequence	Algorithm							
	Deshaker		Warp stabilizer		Video stabilization with robust L1 optimal camera paths		DVS analyzer	
	ITF, dB	Time, s	ITF, dB	Time, s	ITF, dB	Time, s	ITF, dB	Time, s
“SANY0025_xvid.avi”	23.53	1.33	22.7	1.87	22.74	1.34	23.79	0.17
“lf_juggle.avi”	26.65	1.22	24.41	1.64	26.15	1.18	28.06	0.15
“EllenPage_Juggling.avi”	25.61	3.53	26.68	4.53	27.33	3.17	28.58	3.54
“road_cars_krasnoyarsk.avi”	22.31	1.45	21.48	2.15	25.2	1.29	25.91	0.24

moving objects, a luminance, a noise, and the shooting condition. The use of the TSK model provides the increment of ITF estimations up on 3–4 dB or 15–20 %.

The stabilization and temporal results of video sequences from Table 5.1 by existing software tools such as “Deshaker”, “WarpStabilizer”, “Video Stabilization with Robust L1 Optimal Camera Paths”, and our “DVS Analyzer” are located in Table 5.3.

The ITF estimations of the proposed software tool “DVS Analyzer” provides better results (at average 1–3 dB or 5–15 %) with the lower processing time relatively the existing software tools.

## 5.6 Conclusion

The proposed approach for video stabilization of static scenes includes the automatic detection and improvement of blurred frames as well as the LMVs and GMVs estimations using the TSK model in order to separate a camera motion from a motion of moving objects and provide a scene alignment. The development of deblurring method with applied fuzzy logic rules for better motion estimations is discussed in this chapter. All methods and algorithms were realized by the

designed software tool “DVS Analyzer”, v. 2.04. During experiments, the PSNR and the ITF estimations were received for six video sequences with static camera shooting. The ITF estimations increase up on 3–4 dB or 15–20 % relative to the original video sequences.

The development of advanced motion inpainting methods and algorithms for the DVS task and also fast realization of algorithms without essential accuracy fall for pseudo real-time application are the subjects of interest in future researches.

## References

1. Marcenaro, L., Vernazza, G., Regazzoni, C.S.: Image stabilization algorithms for video surveillance applications. *Int. Conf. on Image Process.* **1**, 349–352 (2001). Thessaloniki, Greece
2. Peng, Y.C., Liang, C.K., Chang, H.A., Chen, H.H., Kao, C.J.: Integration of image stabilizer with video codec for digital video cameras. In: *International Symposium on Circuits and Systems*, pp. 4781–4784. Kobe (2005)
3. Rawat, P., Singhai, J.: Review of motion estimation and video stabilization techniques for hand held mobile video. *Int. J. Sig. Image Process.* **2**(2), 159–168 (2011)
4. Tanakian, M.J., Rezaei, M., Mohanna, F.: Digital video stabilization system by adaptive motion vector validation and filtering. In: *International Conference on Communications Engineering*, pp. 165–183. Zahedan (2010)
5. Tanakian, M.J., Rezaei, M., Mohanna, F.: Digital video stabilization system by adaptive fuzzy filtering. In: *Proceedings of the 19th European Signal Processing Conference*, pp. 318–322. Barcelona (2011)
6. Battiato, S., Gallo, G., Puglisi, G., Scellato, S.: Fuzzy-based motion estimation for video stabilization using SIFT interest points. In: *SPIE Electronic Imaging 2009—System Analysis for Digital Photography V EI-7250*, pp. 1–8 (2009)
7. Acharjee, S., Chaudhuri, S.S.: Fuzzy logic based three step search algorithm for motion vector estimation. *Int. J. Image Graph. Sig. Process.* **2**, 37–43 (2012)
8. Gullu, M.K., Erturk, S.: Membership function adaptive fuzzy filter for image sequence stabilization. *IEEE Trans. Consum. Electron.* **50**(1), 1–7 (2004)
9. Sugeno, M.: *Industrial applications of fuzzy control*. Elsevier Science Inc., New York (1985)
10. Kyriakoulis, N., Gasteratos, A.A.: Recursive fuzzy system for efficient digital image stabilization. *Advan. in Fuzzy Syst.* **2008**, 1–8 (2008)
11. Shen, Y., Guturu, P., Damarla, T., Buckles, B.P., Namuduri, K.R.: Video stabilization using principal component analysis and scale invariant feature transform in particle filter framework. *IEEE Trans. on Consum. Electron.* **55**(3), 1714–1721 (2009)
12. Tsai, D., Lai, S.: Defect detection in periodically patterned surfaces using independent component analysis. *Pattern Recogn.* **41**(9), 2812–2832 (2008)
13. Kim, N., Lee, H., Lee, J.: Probabilistic global motion estimation based on Laplacian two-bit plane matching for fast digital image stabilization. *EURASIP J. Adv. Sig. Process.* pp. 1–10 (2008)
14. Pun, C.M., Lee, M.C.: Log-polar wavelet energy signatures for rotation and scale invariant texture classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 590–603 (2003)
15. Shakoor, M.H., Moattari, M.: Statistical digital image stabilization. *J. Eng. Technol. Res.* **3**(5), 161–167 (2011)
16. Cho, S., Wang, J.: Video deblurring for hand-held cameras using patch-based synthesis. *ACM Trans. Graph. (SIGGRAPH 2012)* **31**(4), 1–64 (2012)
17. Cho, S., Lee, S.: Fast motion deblurring. *ACM Trans. Graph.* **28**(5), 1–8 (2009)
18. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Trans. Graph.* **27**(3), 1–10 (2008)

19. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. *Int. J. Comput. Vis.* **98**(2), 168–186 (2012)
20. Gupta A., Joshi N., Zitnick C.L., Cohen M., Curless B.: Single image deblurring using motion density functions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *Computer Vision—ECCV 2010*, LNCS 6311, Part 1, pp. 171–184. Springer, Heidelberg (2010)
21. Cai, J., Walker, R.: Robust video stabilization algorithm using feature point selection and delta optical flow. *IET Comput. Vis.* **3**(4), 176–188 (2009)
22. Cho S., Matsushita Y., Lee S.: Removing non-uniform motion blur from images. In: *11th International Conference on Computer Vision*, pp. 1–8. Rio de Janeiro (2007)
23. Matsushita, Y., Ofek, E., Ge, W., Tang, X., Shum, H.Y.: Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(7), 1150–1163 (2006)
24. Favorskaya, M., Buryachenko, V.: Video stabilization of static scenes based on robust detectors and fuzzy logic. *Front. Artif. Intell. Appl.* **254**, 11–20 (2013)