

Hindawi Publishing Corporation  
Advances in Multimedia  
Volume 2013, Article ID 651650, 8 pages  
<http://dx.doi.org/10.1155/2013/651650>

## Research Article

# An Integrated New Scheme for Digital Video Stabilization

W. Xu,<sup>1,2</sup> X. Lai,<sup>1</sup> D. Xu,<sup>3</sup> and N. A. Tsofigkas<sup>3</sup>

<sup>1</sup> Institute of Electronic CAD, Xidian University, Xi'an 710071, China

<sup>2</sup> Xi'an Aeronautical University, Xi'an 710077, China

<sup>3</sup> School of Science & Engineering, Teesside University, Middlesbrough TS1 3BA, UK

Correspondence should be addressed to D. Xu; [d.xu@tees.ac.uk](mailto:d.xu@tees.ac.uk)

Received 24 April 2013; Revised 7 November 2013; Accepted 14 November 2013

Academic Editor: Jianping Fan

Copyright © 2013 W. Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In many digital video applications, video sequences suffer from jerky movements between successive frames. In this paper, an integrated general-purpose stabilization method is proposed, which extracts the information from successive frames and removes the translation and rotation motions that result in undesirable effects. The scheme proposed starts with computation of the optical flow between consecutive video frames and an affine motion model is adopted in conjunction with the optical flow field obtained to estimate objects or camera motions using the Horn-Schunck algorithm. The estimated motion vectors are then used by a model-fitting filter to stabilize and smooth video sequences. Experimental results demonstrate that the proposed scheme is efficient due to its simplicity and provides good visual quality in terms of the global transformation fidelity measured by the peak-signal-noise-ratio.

## 1. Introduction

Video captured by cameras often suffers from unwanted jittering motions. In general, this problem is usually dealt with by means of compensation for image motions. Most video stabilization algorithms presented in the recent literature try to remove the image motions by either totally or partially compensating for all motions caused by camera rotations or vibrations [1–9]; therefore the resultant background remains motionless. The motion models described in [1, 2] proposed a pyramid structure to compute the motion vectors with an affine motion model representing rotational and translational camera motions. Hansen et al. [3] described an image stabilization scheme which uses a multiresolution, iterative process to calculate the affine motion parameters between levels of Laplacian pyramid images. The parameters obtained through the refinement process achieve the desired accuracy. The method presented in [4] used a probabilistic model with a Kalman filter to reduce the motion noises and to obtain stabilized camera motions. Chang et al. [5] used the optical flow between consecutive frames based on the modification of the method in [6] to estimate the camera motions by fitting a simplified affine motion model. Tsubaki et al. [7] developed a method that uses two threshold parameters to describe

the velocity and the frequency of oscillations of unstable video sequences. More recently, Zhang et al. [8] proposed a 3D perspective camera model based method, which works well in situations where significant depth variations exist in the scenes and the camera undergoes large translational movement. The technique developed in [9] adopted a spatially and temporally optimized approach to achieve high-quality camera motion on videos where 3D reconstruction is difficult or long feature trajectories are not available. The technique formulates stabilization as a spatial-temporal optimization problem that finds smooth feature trajectories and avoids visual distortion.

In this paper, an integrated video stabilization scheme is proposed, which primarily has two objectives. First of all, rather than developing novel and complicated individual algorithms, it aims to simplify the stabilization process by integrating the well-researched techniques, such as motion estimation, motion modeling, and motion compensation, into a new single framework that is of modular nature and can reduce the complexity for implementation, particularly in hardware. Secondly, the scheme aims to provide better performance in terms of the global transformation fidelity (a typical measure of stabilization performance), compared to other existing methods. This is achieved by combining optical

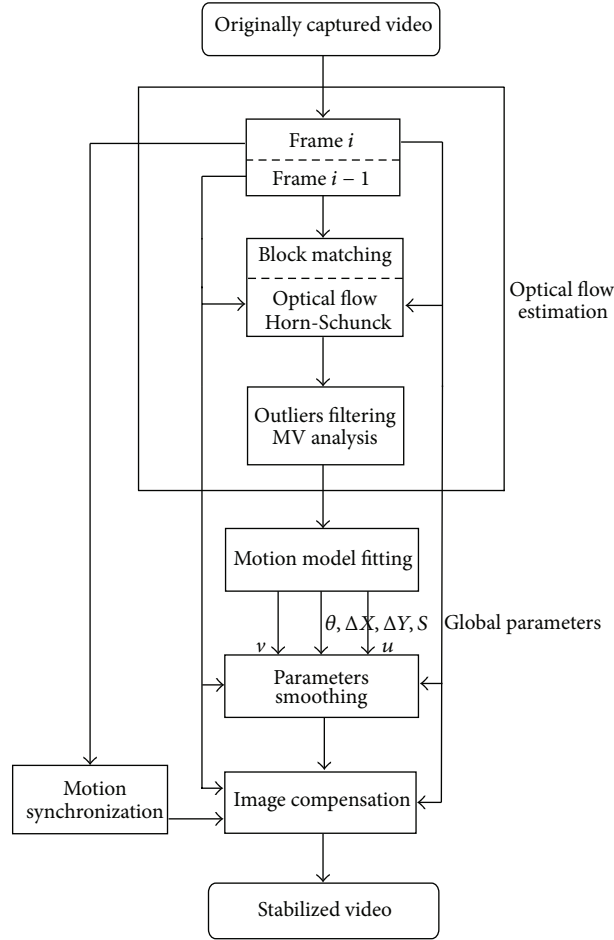


FIGURE 1: Flowchart of the video stabilization scheme.

flow estimation with motion models to increase accuracy of estimation. The scheme is based on estimating the motion field between consecutive frames using the Horn-Schunck algorithm [10]. An iterative process based on a coarse-to-fine technique is adopted here. The motion vectors are firstly estimated using the block matching method between two consecutive fields and then the dense motion field is estimated using the motion vectors and the Horn-Schunck algorithm. By fitting an affine motion model, the motion parameters of the model are computed and smoothed. Thus, by analyzing the directions of motion vectors and their standard deviations as well as using previously stabilized frame or a reference frame, the image motions caused by three-dimensional rotation and translation can be determined and the current video frame can be stabilized.

The rest of this paper is organized as follows. In the next section, we present an overview of the proposed video stabilization scheme. Sections 3, 4, 5, and 6 describe in detail key components of the scheme, namely, optical flow field estimation, motion model fitting, motion parameter smoothing, and motion compensation. The experimental and simulation results of the proposed method are presented in Section 7. Finally, the conclusions are drawn in Section 8.

## 2. Overview of the Video Stabilization Scheme

The flowchart of the proposed stabilization scheme is shown in Figure 1, which integrates four key components: optical flow field estimation, motion model fitting, motion parameters smoothing, and motion compensation. Each of these components will be analytically presented in detail in the following sections.

## 3. Optical Flow Estimation Technique

The accuracy of the stabilization scheme mainly depends on the motion vectors produced during the interframe motion estimation. Here, a coarse-to-fine technique is used to perform block correlation, initially at a coarse scale, and then to interpolate the resulting estimates before they pass through iterations of Horn and Schunck's optical flow algorithm. Optical flow is an approximation of the local image motion based upon local derivatives in a given sequence of images. That is, in two dimensions, it specifies how much each image pixel moves between adjacent images, while, in three dimensions, it specifies how much each volume voxel moves between adjacent volumes.

To estimate optical flow of any pixel  $(x, y)$  in an image, we use the “intensity constancy” assumption which states that the intensity of any pixel on any object in an image remains constant with time; that is,

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t). \quad (1)$$

Assuming small motions between consecutive frames (small  $\delta x$  and  $\delta y$ ), we can perform a first-order Taylor series expansion on  $I(x, y, t)$  in (1) to obtain

$$\begin{aligned} I(x + \delta x, y + \delta y, t + \delta t) \\ = I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + \text{H.O.T.} \end{aligned} \quad (2)$$

H.O.T. are the higher order terms of the Taylor series, which we assume to be small and can safely be ignored. Using (1) and (2), we can obtain

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0 \quad (3)$$

or

$$I_x u + I_y v + I_t = 0, \quad (4)$$

where  $I_x = \partial I / \partial x$ ,  $I_y = \partial I / \partial y$ ,  $I_t = \partial I / \partial t$ ,  $u = \delta x / \delta t$ , and  $v = \delta y / \delta t$ .

Equation (4) has two variables,  $u$  and  $v$ , which means that, for an image with  $N$  pixels, there will be  $N$  equations with  $2N$  variables. Hence additional constraints are required to solve these equations. Horn and Schunck proposed to use the smoothness constraint; that is, to find  $u$ ,  $v$ , and their derivatives, we minimize the following energy function:

$$\begin{aligned} I(u, v) \\ = \iint_{\Omega} \left[ (I_x u + I_y v + I_t)^2 + \lambda^2 (u_x^2 + u_y^2 + v_x^2 + v_y^2) \right] dx dy, \end{aligned} \quad (5)$$

where  $\lambda$  controls the weight given to the smoothness constraint and  $\Omega$  denotes the image domain. Also we assume that  $u$  and  $v$  are zero at the boundaries of the image domain. The minimization of (5) is achieved using calculus of variations and the approximation of the Laplacian:

$$\begin{aligned} \nabla^2 u &\approx k (\bar{u}_{i,j,k} - u_{i,j,k}), \\ \nabla^2 v &\approx k (\bar{v}_{i,j,k} - v_{i,j,k}). \end{aligned} \quad (6)$$

The derivatives of brightness are estimated from the discrete set of image brightness measurements as follows:

$$\begin{aligned} I_x &= \frac{1}{4} (I_{i,j+1,k} - I_{i,j,k} + I_{i+1,j+1,k} \\ &\quad - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j,k+1} \\ &\quad + I_{i+1,j+1,k+1} - I_{i+1,j,k+1}), \end{aligned}$$

$$\begin{aligned} I_y &= \frac{1}{4} (I_{i+1,j,k} - I_{i,j,k} + I_{i+1,j+1,k} \\ &\quad - I_{i,j+1,k} + I_{i+1,j,k+1} - I_{i,j,k+1} \\ &\quad + I_{i+1,j+1,k+1} - I_{i,j+1,k+1}), \\ I_t &= \frac{1}{4} (I_{i,j,k+1} - I_{i,j,k} + I_{i+1,j,k+1} \\ &\quad - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j+1,k} \\ &\quad + I_{i+1,j+1,k+1} - I_{i+1,j+1,k}), \end{aligned} \quad (7)$$

where  $i$  and  $j$  are the grid space intervals and  $k$  is the image frame sampling period. The local averages  $u$  and  $v$  are defined as follows:

$$\begin{aligned} \bar{u}_{i,j,k} &= \frac{1}{6} (u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}) \\ &\quad + \frac{1}{12} (u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}), \\ \bar{v}_{i,j,k} &= \frac{1}{6} (v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}) \\ &\quad + \frac{1}{12} (v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k}). \end{aligned} \quad (8)$$

We can compute a new set of velocity estimates  $(u^{n+1}, v^{n+1})$  per frame from the estimated derivatives and the average of the previous velocity estimates  $(u^n, v^n)$  by

$$\begin{aligned} u_{i,j}^{n+1} &= \bar{u}_{i,j}^n - \alpha I_x, \\ v_{i,j}^{n+1} &= \bar{v}_{i,j}^n - \alpha I_y, \\ \alpha &= \frac{I_x \bar{u}_{i,j}^n + I_y \bar{v}_{i,j}^n + I_t}{\lambda^2 + I_x^2 + I_y^2}. \end{aligned} \quad (9)$$

At the iteration  $n = 0$ , the initial values of  $u$  and  $v$  are set equal to zero.

In motion estimation, there are occasions that the motion vectors produced fall outside normal values. When a motion vector is above a certain value, it is characterized as an outlier. The above method is very sensitive to outliers; that is, it is prone to produce outliers or unexpected data. Therefore, an alternative value has to be considered to substitute these outliers. Here, the median value of motion vectors is adopted. This is because, among geometric mean, harmonic mean, standard deviation, median and trim-mean, all of which have been applied and tested, the median and trim-mean are found to be the most robust, that is, resistant to outliers.

#### 4. Motion Model Fitting

A camera projects a three-dimensional world point onto a two-dimensional image point. The motion of the camera may be regarded as a single motion such as rotation, translation, or

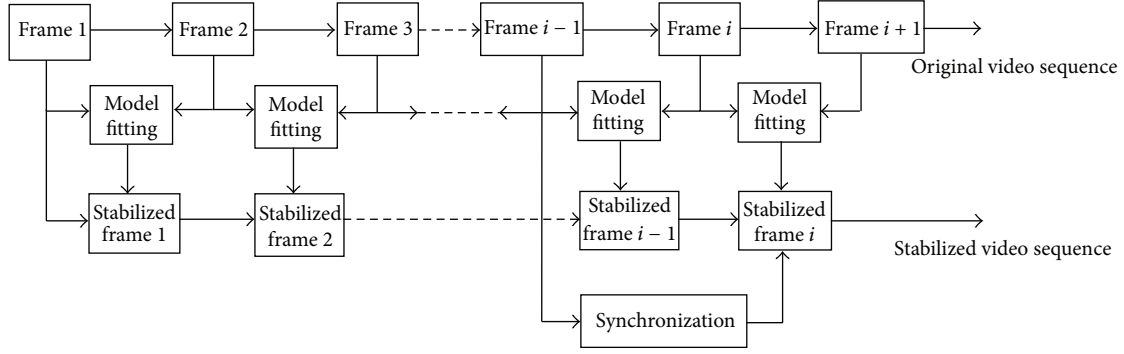


FIGURE 2: Compensation process to stabilize video sequence.

zoom or a combination of any two or three of these motions. Such camera motion can be well categorized by a set of parameters. In our case, the first frame of a video sequence is used to define the reference coordinate system, and a two-dimensional affine model is used to estimate a parametric form describing the displacement of the video content between consecutive frames by identifying the correspondence between local invariant features. The affine model was employed since it is more resilient to noisy data and it can represent all the basic camera motions which often occur in video applications. If we denote a pixel position in the first frame by  $(x, y)$  and the corresponding position in the second frame by  $(x', y')$ , the two-dimensional affine motion model can be formulated as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_0 \\ a_3 \end{pmatrix} + \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \quad (10)$$

where the motion parameters  $a_1, a_2$  control the scaling and rotation ( $a_1 = \cos \theta$  and  $a_2 = \sin \theta$  if there is only rotation;  $\theta$  is rotation angle) and the parameters  $a_0$  and  $a_3$  correspond to the horizontal and vertical translations.

Assuming that we have  $N$  motion vectors, which correspond to the image pixels  $(x_i, y_i)$  and  $(x'_i, y'_i)$  ( $i = 1, 2, 3, \dots, N$ ) between two consecutive video frames, we can estimate the simplified affine motion between the two frames from the  $N$  motion vectors by solving the following overconstrained linear equation:

$$\begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & -y_N & 1 & 0 \\ y_N & x_N & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_0 & a_3 \end{bmatrix}^T = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ x'_N \\ y'_N \end{bmatrix} = \begin{bmatrix} x_1 + u_1 \\ y_1 + v_1 \\ x_2 + u_2 \\ y_2 + v_2 \\ \vdots \\ x_N + u_N \\ y_N + v_N \end{bmatrix}. \quad (11)$$

A                      X                      Y

The affine motion parameters are obtained through solving this linear equation by the least-square solution. Equation  $\mathbf{AX} = \mathbf{Y}$  has least square solution if  $\mathbf{A}^t \mathbf{A}$  is invertible and

$\text{rank}(\mathbf{A}) = 4$ . In this case, the least square solution  $\mathbf{X}$  is given by  $\mathbf{X} = (\mathbf{A}^t \mathbf{A})^{-1} \mathbf{A}^t \mathbf{Y}$ .

## 5. Motion Parameters Smoothing

In order to produce high quality stabilized video sequences, the motion parameters obtained need to be smoothed. This can be achieved by space-domain filtering. Different types of filters have been applied and tested. These include the recursive Kalman filtering which removes camera vibrations, the moving average filter that smoothes data by replacing each data with the average of the neighboring data defined within a span, and the locally weighted scatterplot smoothing which uses weighted linear regression to smooth data. In our scheme, the Savitzky-Golay filter [11, 12] is used to process the originally estimated affine global motion parameters as it is a generalized moving average filter with the properties of simplicity and efficiency for implementation.

## 6. Global Motion Compensation

Motion compensation is performed frame by frame using previously stabilized frames (apart from the first frame) and their corresponding global smoothed parameters; that is, the first stabilized frame is obtained by compensating the first original frame with its corresponding smoothed affine motion parameters; the second stabilized frame is achieved by compensating the first stabilized frame with its corresponding smoothed affine motion parameters, and so forth. The block diagram of this compensation process is shown in Figure 2. Due to utilization of the first original frame (rather than a previously stabilized one) at the very beginning of the process, an error will be produced and propagated to the subsequent frames. In order to mitigate this effect, synchronization is performed to control the error as follows. For the frame  $i$ , the unsmoothed motion parameters are compared with the smoothed motion parameters. If the result of the comparison is less than a threshold (synchronization distance threshold), the original frame  $i-1$  together with the corresponding smoothed motion parameters are used to obtain the stabilized frame  $i$ , that is, synchronizing the stabilized frame  $i$  with the original frame  $i-1$ ; otherwise the stabilized frame  $i-1$  with the corresponding smoothed motion parameters is

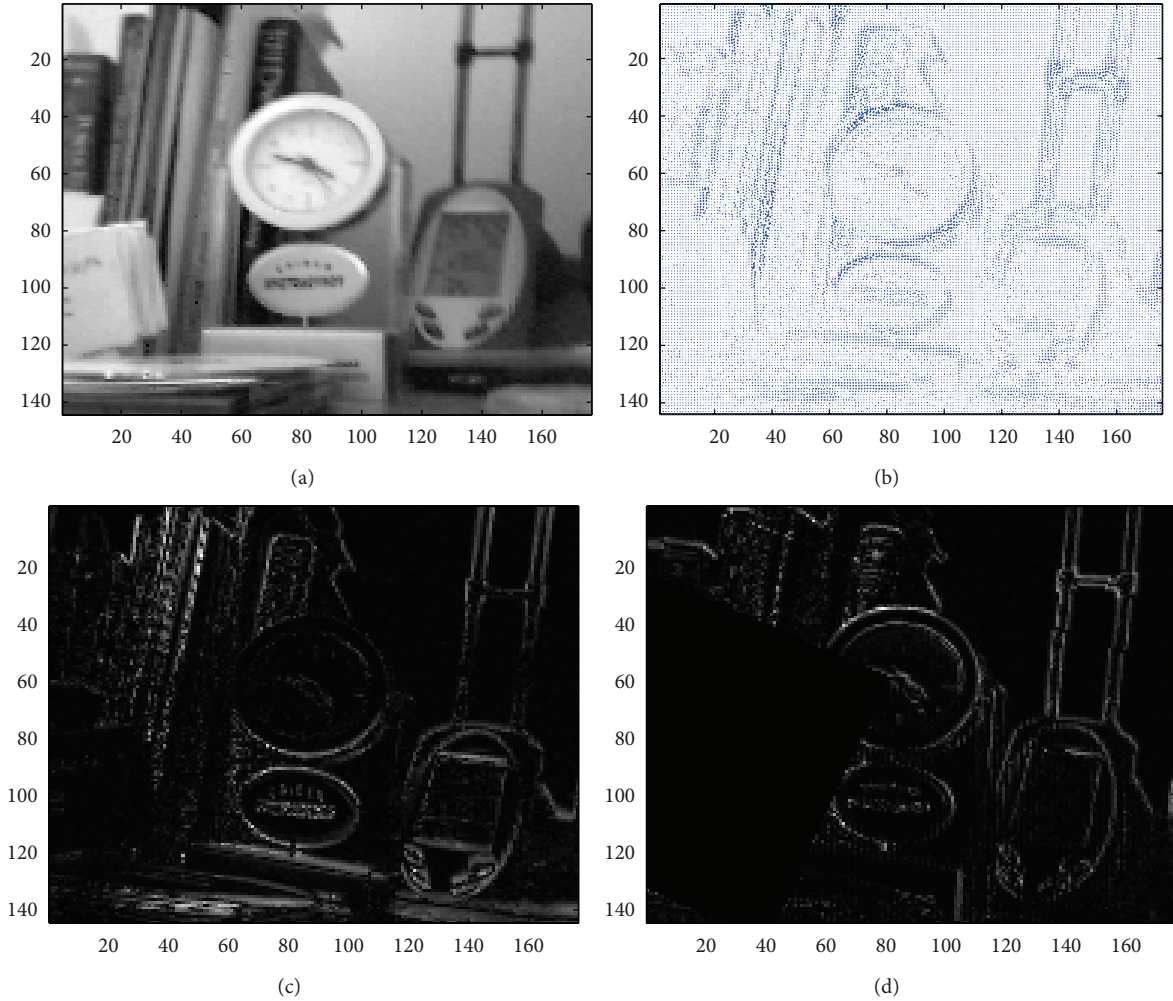


FIGURE 3: Example of stabilization results from the stabilization scheme proposed: (a) original frame No. 14 of the video sequence “My Office”; (b) dense optical flow field-estimated motion vectors between frames 14 and 15; (c) difference between original frames 14 and 15; (d) difference between stabilized frames 14 and 15.

used to obtain the stabilized frame  $i$ . The higher the threshold is, the more synchronized the stabilized video sequence would be with the original video sequence, therefore reducing errors. In order to guarantee quality of the stabilized output video, regardless of the error control described above, a synchronization frame is enforced every 30 frames.

## 7. Simulation Results

In order to evaluate the effectiveness and performance of the stabilization scheme proposed, the simulations are carried out using a range of the QCIF format (176 pixels by 144 lines) video sequences captured. Figure 3 shows the stabilization results from the captured video sequence “My Office.” Figure 3(a) shows an original frame (no. 14) from the video sequence. The frame clearly contains the “tremor effect,” which is deliberately introduced into the video sequence. Figure 3(b) shows the optical flow field, which is estimated from frames 14 and 15. The random vectors detected, as shown in the figure, are due to zooming in/out effects produced during

the video recording. Since it is very difficult to visually distinguish two consecutive frames, here we compare the difference between two original frames (numbers 14 and 15) to the difference between two corresponding stabilized frames, as shown in Figures 3(c) and 3(d). The results of another experiment from the video sequence “Jerky” are shown in Figure 4. Similarly, the frames 14 and 15 of the sequence are used here. From these experiments, it is obvious that the stabilized frames in Figures 3(d) and 4(d) have much less motion (white pixels/regions) than the original frames in Figures 3(c) and 4(c) do. This demonstrates that, after the stabilization process, a significant amount of the undesirable movements have been compensated.

Since dynamic processes, such as stabilization, cannot be illustrated with still images, we present and compare in Figure 5 the three motion parameters (rotation, horizontal, and vertical displacements) of the first 80 frames of the video sequence “My Office.” The comparisons between original (blue curve) and smoothed (red curve) motion parameters show that all three motion parameters have been smoothed

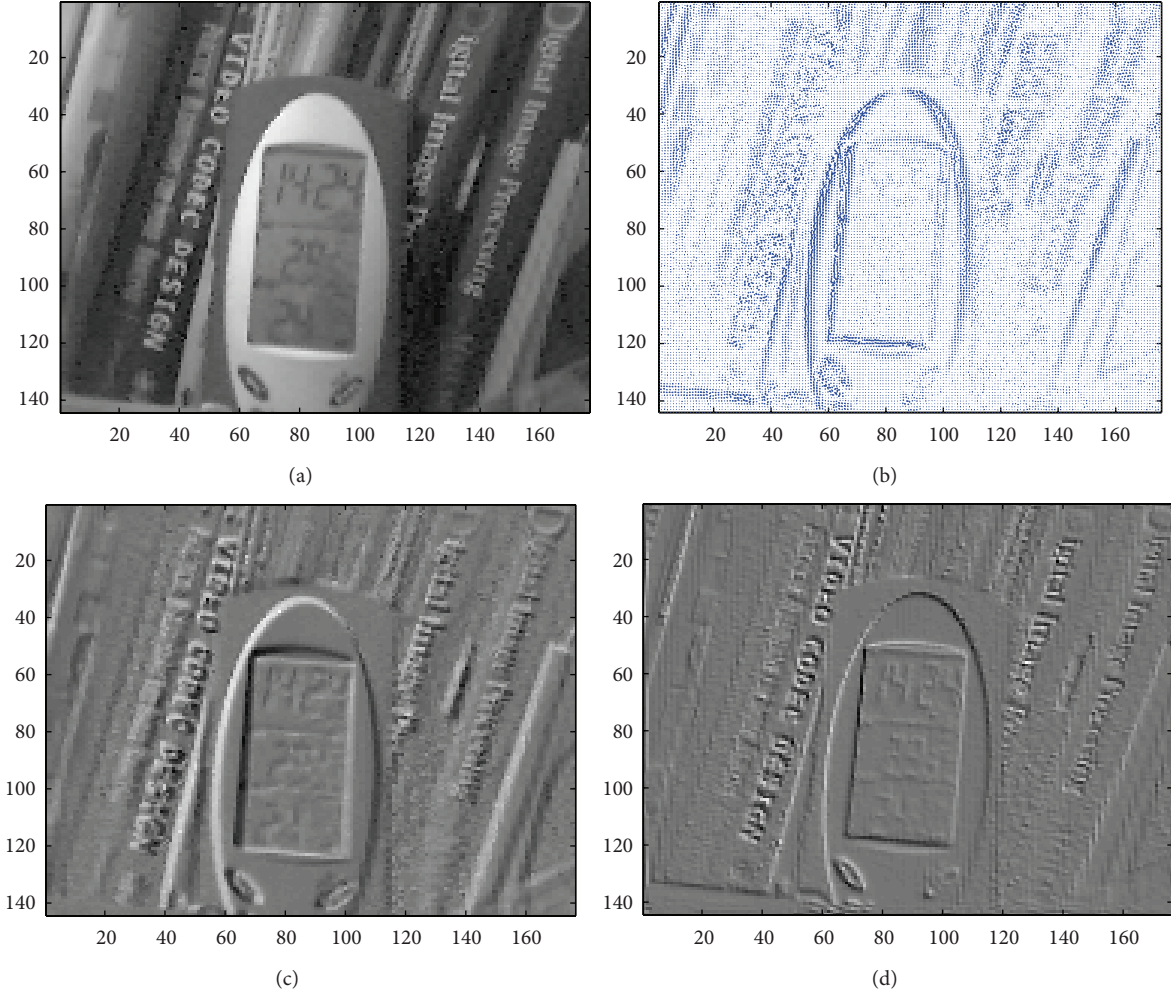


FIGURE 4: Experimental results from the video sequence “Jerky”: (a) original frame no. 14; (b) dense optical flow field-estimated motion vectors between frames 14 and 15; (c) difference between original frames 14 and 15; (d) difference between stabilized frames 14 and 15.

for the length of the video sequence, therefore reducing the unwanted movements captured during the generation of the video. It is also shown that the parameter smoothing process has been improved with synchronization. Without this synchronization (green curve), accumulated estimation errors will increase significantly after a certain number of frames. The difference between the red and green curves indicates clearly that the parameter smoothing process is synchronized at frames 7, 13, 25, 41, 57, and 73 in order to correct the accumulated estimation errors, which occur at these frame locations on the green curves.

In order to objectively and quantitatively evaluate the performance of the scheme proposed, we use global transformation fidelity (GTF) [13] as a measure of how well stabilization compensates the motion of a camera; that is, how precisely the motion model fits the actual camera motion. Here, the Peak-Signal-Noise-Ratio (PSNR) between two frames is used to measure the GTF, which is defined as

$$\text{PSNR}(f_2, f_1) = 10 \log \frac{255^2}{\text{MSE}(f_2, f_1)}, \quad (12)$$

where MSE (mean squared error) measures the average difference between the two frames  $f_1$  and  $f_2$ . Figure 6 shows the performance comparison in terms of PSNR between the proposed stabilization scheme and the well-known Gray-Coded Bit-Plane Matching based stabilization method [14, 15], which uses simple Gray-coding and has low computational load for hardware implementation, therefore being of equivalent complexity to that of the proposed scheme. The calculations of PSNR in the figure make use of the first 50 frames of “My Office” sequence, which contains the anticlockwise camera rotation of 90 degrees. The GTF (PSNR) is calculated between the reference frame (the first frame in this case) and the currently stabilized frame. It can be observed from the GTF curves that the proposed stabilization procedure performs significantly better than the Gray-Coded Bit-Plane Matching during the first 10 frames which correspond to the rotation part of the sequence. This result is anticipated because the Gray-Coded Bit-Plane Matching method does not compensate for rotation very well. The GTF of the proposed scheme drops from frame to frame since each following frame has less overlap with the reference frame. After

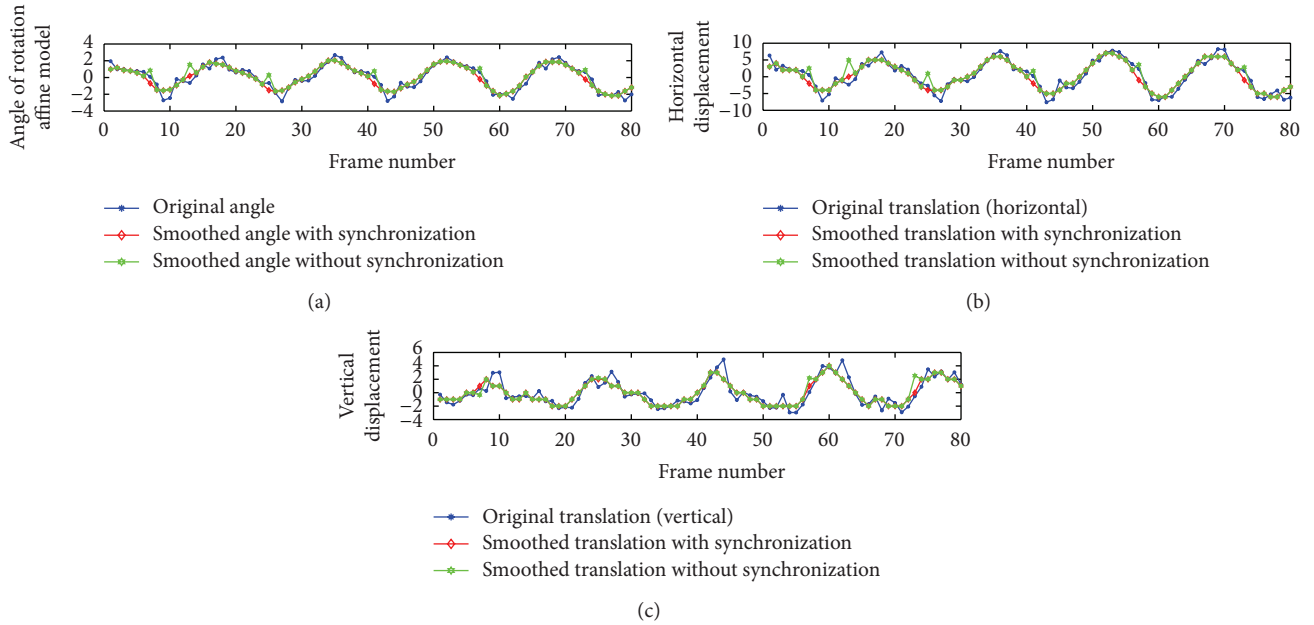


FIGURE 5: Original and smoothed motion parameters of the video “My Office.”

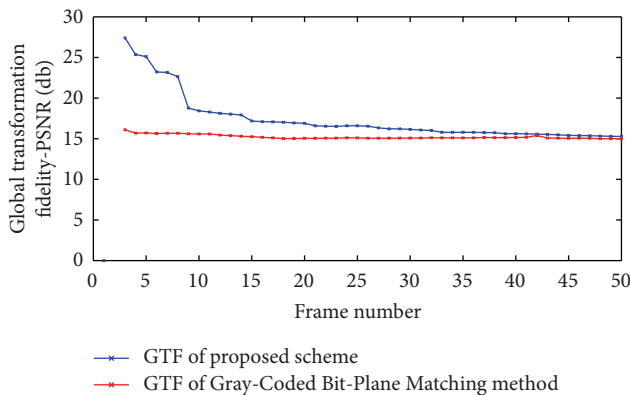


FIGURE 6: GTF of the video sequence “My Office,” measured by PSNR.

about 40 frames, the sequence almost does not overlap with the reference frame.

## 8. Conclusions

This paper presents a general-purpose video stabilization scheme, aiming at a simple and effective solution for a wide range of video-based applications. The scheme features integration of optical flow and motion model based motion estimation, space-domain filtering, and motion compensation, thus offering an efficient computation method for video stabilization. It is successfully implemented in MATLAB. The simulation result shows that the scheme is effective for a broad range of real-time applications. Compared to other video stabilization methods, it has the advantages of simplicity and robustness while maintaining better or comparable performance in terms of the global transformation fidelity measured by PSNR.

## References

- [1] C. Morimoto and R. Chellapa, “Automatic digital image stabilization,” in *Proceedings of the IEEE International Conference on Pattern Recognition*, pp. 660–665, Vienna, Austria, 1996.
- [2] C. Morimoto and R. Chellapa, “Fast electronic digital image stabilization for off-road navigation,” *Real-Time Imaging*, vol. 2, no. 5, pp. 285–296, 1996.
- [3] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, “Real-time scene stabilization and mosaicking,” in *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*, pp. 54–62, December 1994.
- [4] A. Litvin, J. Konrad, and W. C. Karl, “Probabilistic video stabilization using Kalman filtering and mosaicking,” in *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging, Image and Video Communications and Processing*, Proceedings of SPIE, pp. 663–674, January 2003.
- [5] H.-C. Chang, S.-H. Lai, and K.-R. Lu, “A robust and efficient video stabilization algorithm,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, pp. 29–32, Taipei, Taiwan, June 2004.
- [6] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI '81)*, pp. 674–679, Vancouver, Canada, 1981.
- [7] I. Tsubaki, T. Morita, T. Saito, and K. Aizawa, “An adaptive video stabilization method for reducing visually induced motion sickness,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP '05)*, pp. 497–500, Genoa, Italy, September 2005.
- [8] G. Zhang, W. Hua, X. Qin, Y. Shao, and H. Bao, “Video stabilization based on a 3D perspective camera model,” *Visual Computer*, vol. 25, no. 11, pp. 997–1008, 2009.
- [9] Y. S. Wang, F. Liu, P. S. Hsu, and T. Y. Lee, “Spatially and temporally optimized video stabilization,” *IEEE Transactions on*

- Visualization and Computer Graphics*, vol. 19, no. 8, pp. 1354–1361, 2013.
- [10] B. K. P. Horn and B. G. Schunck :, “Determining optical flow,” Massachusetts Institute of Technology, Artificial Intelligence Laboratory, A.I Memo no. 572, 1980.
  - [11] A. Savitzky and M. J. E. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.
  - [12] N. A. Tsoligkas, D. Xu, I. French, and Y. Luo, “A motion model based video stabilisation algorithm,” in *Proceedings of the World Automation Congress (WAC '06)*, Budapest, Hungary, June 2006.
  - [13] C. Morimoto and R. Chellappa, “Evaluation of image stabilization algorithms,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '98)*, vol. 5, pp. 2789–2792, Seattle, Wash, USA, May 1998.
  - [14] 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.
  - [15] A. Çelebi, O. Akbulut, O. Urhan, and S. Ertürk, “Truncated gray-coded bit-plane matching based motion estimation and its hardware architecture,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1530–1536, 2009.