Journal of Shanghai University (English Edition), 2005, 9(1): 46-51
Article ID: 1007-6417(2005)01-0046-06

Robust Video Stabilization Based on Motion Vectors

SONG Li(宋利), ZHOU Yuan-hua(周源华), ZHOU Jun(周军)
Institute of Image Communication and Information Processing, Shanghai Jiaotong University, Shanghai 200030, P. R. China

Abstract This paper proposes a new robust video stabilization algorithm to remove unwanted vibrations in video sequences. A complete theoretical analysis is first established for video stabilization, providing a basis for new stabilization algorithm. Secondly, a new robust global motion estimation (GME) algorithm is proposed. Different from classic methods, the GME algorithm is based on spatial-temporal filtered motion vectors computed by block-matching methods. In addition, effective schemes are employed in correction phase to prevent boundary artifacts and error accumulation. Experiments show that the proposed algorithm has satisfactory stabilization effects while maintaining good tradeoff between speed and precision.

Key words video stabilization, global motion, robust estimation.

1 Introduction

The goal of video stabilization is to remove unwanted motions from a moving video sequence. It is important in many vision tasks such as tele-operation, robot navigation, ego-motion recovery, scene modeling, video compression and detection of independent moving objects. Many methods for video stabilization have been reported over the past few years. They are largely divided into two categories: featured-based method[1] and intensity-based method[2]. The first type of method tracks a set of features through the sequence, and uses their motions to estimate stabilizing warping. These methods have good performance when scenes are relatively still or slowly changing. However, feature-based methods face difficulty when scenes with moving objects. The second type of method tracks image intensities of single region dynamically in a multi-resolution way; these methods have better precision since they estimate global motion based on every pixel in the overlapping area of consecutive sequences. The disadvantage is that these methods need complicated motion models to deal with different objects in the scene and require a large amount of computation.

Received Sep. 28, 2003; Revised Feb. 16, 2004
Project supported by the National High-Technology Research and Development Program of China (Grant No. 863—2002AA103087)
SONG Li, Ph. D. Candidate, E-mail: songli@qantosft.com;
ZHOU Yuan-hua, Prof., E-mail: yuanhuazhou@cdtv.org.cn

In this paper, we first establish a theory of video stabilization under affine motion models about camera, which clearly shows how global motion parameters (raw and smoothing) are used to stabilize original dithering video sequences. In order to improve computing efficiency without loss of precision, we propose a new global motion estimation algorithm, which computes motion vectors through a block-matching method and then introduces a simplified M robust method to estimate global parameters after filtering raw motion vectors in the spatial and temporal domain. After obtaining global motion parameters of consecutive frame pairs, we employ median filtering to get "real" or desired global motion parameters. In the phase of motion correction, we employ mosaic-based edge region compensation scheme to reduce the artifacts of undefined area on the boarder; at the same time we design a "re-synchronization" scheme to deal with error accumulation. Experiments show that the proposed algorithm can achieve a good tradeoff between precision and robustness. In addition, it is convenient to be extended to the compressed domain.

2 Theory of Video Stabilization

For video applications, it is appropriate to describe the camera's global motion by a 2D transformation. There are several 2D transformation models with increasing complexity: two-parameter translation models, four-parameter similarity models, six-parameter affine models and eight-parameter projective models[3]. The more the complex motion model, the more free-

dom it allows in deforming observed 2D scenes. However, with increasing complexity, computation complexity, sensitivity to numerical errors, model failures and noise also increase. In this paper, we use the affine transformation model for its good tradeoff between complexity and computing speed. Under an affine transformation, corresponding pixels X_n and X_{n-1} in image I_n and I_{n-1} are related by:

$$X_n = A_n X_{n-1} + T_n = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} X_{n-1} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \quad (1)$$

where t_1 and t_2 describe the translation, a_1 and a_4 describe the scaling, and a_2 and a_3 describe the shear of the object. Taking the first video frame as a reference and cascade the above equation frame by frame, we can obtain:

$$X_n = A_n X_{n-1} + T_n = \cdots = A_n^r X_1 + T_n^r,$$

$$\text{where } A_n^r = \left(\prod_{i=1}^{n} A_i \right) X_1,$$

$$T_n^r = \sum_{i=1}^{n-1} \left(\prod_{k=i+1}^{k=n} A_k \right) T_i + T_n. \quad (2)$$

This equation indicates that frame n and the reference frame can be related by the cumulative transform parameters (A_n^r, T_n^r). Suppose the intentional cumulative transformation is (A_n^s, T_n^s), we can get a similar equation as follows:

$$X_n^s = A_n^s X_1 + T_n^s. \quad (3)$$

From Eqs. (2) and (3), we can get a new relation:

$$X_n^s = A_n^c X_n + T_n^c,$$

$$\text{where } A_n^c = A_n^s (A_n^r)^{-1},$$

$$T_n^c = T_n^s - A_n^s (A_n^r)^{-1} T_n^r. \quad (4)$$

To stabilize the video, compensation of each frame for unwanted transformation must be employed. From the above equation, we can get a new corrected frame I_n^c by computing

$$I_n^c(X) = I_n((A_n^c)^{-1} X - (A_n^c)^{-1} T_n^c). \quad (5)$$

Image values at non-integer locations in Eq. (5) are obtained with bilinear interpolation.

3 Framework of the Proposed Algorithm

From the theory of video stabilization described in the previous section, the key part of the whole stabilization processing lies in global motion estimation of two adjoining frames. Here we propose the algorithm framework as illustrated in Fig. 1.

The original streams are first processed using a blocked-based motion estimation (BME) module. Since not all motion vectors from BME contribute to global motion estimation, the original motion vectors are filtered with a spatial-temporal filter (STF) module to get highly reliable motion vectors. Afterwards, the filtered motion vectors are used to global motion estimation (GME), where we introduce simplified M-estimation to further improve robustness. When obtaining the global motion parameters between consecutive frames, we get smooth global motion parameters by time average filtering (TAF). Finally, we use computed global motion parameters to correct motion (CM) of original stream and get stabilized stream.

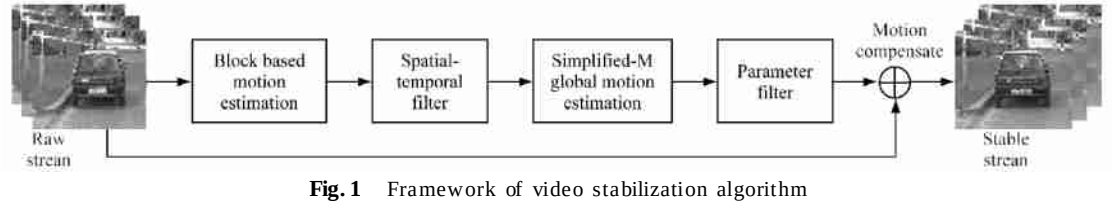


Fig. 1 Framework of video stabilization algorithm

3.1 Block-based motion estimation

Motion estimation has low computation complexity and easy to implement. It has been widely adopted by video coding standards such as MPEG-1, MPEG-2, MPEG-4, H.263, and H.26L. These standards employ the MAD of a macro block as their matching cost function. However, MAD value of the best match depends on the illumination, thus choosing the variance of the MAD as cost function is more reliable. So the new cost function is defined as follows:

$$C = \sum_{i \in M, j \in M} (D_{i,j} - \mu)^2,$$

$$D_{i,j} = | I_1(i,j) - I_2(i + d_1, j + d_2) | ,$$
$$\mu = \sum_{i \in M, j \in M} D(d_1, d_2) / M \times N , \quad (6)$$

where $I_i (i = 1,2)$ is image intensity, the block size is $M \times N$. In the present implementation, $M = N = 8$ or 16. $d_i (i = 1,2)$ is search step with a search area of ± 15 pixels. After motion estimation, we get a series of mean distributed motion vectors. Such motion vectors commonly contain noise and outlier. In video compression, remaining variances are also sent to the receiver, so it only affects the coding rate. However, since our purpose is to obtain global motion information, measures should be taken to filter such motion vectors.

3.2 The spatial-temporal filtering

As motion vectors vary greatly with video contents, it is hard to give an appropriate model to describe noise distribution. However, since error possibility of motion vectors is high in cases of low-texture or uniform areas, we first filter motion vectors corresponding to these areas.

A convolution mask is operated on each block, and the number of pixels with a gradient above a given threshold is counted as follows:

$$S = \begin{cases} 0 & | G_x | < d \cup (| G_y | < d) , \\ 1 & \text{other.} \end{cases} \quad (7)$$

The blocks that have too few edge pixels are deemed low-textured area and therefore excluded in further consideration. This method is fairly robust, and effectively acts as a binary mask applied to the motion field. In fact, if the above method is implemented before motion estimation, the number of blocks used to compute motion vectors will be greatly reduced. However, some streams such as that of MPEG itself include motion vectors information, which can be obtained directly together with the similar binary mask by analyzing DC coefficients[6].

If scenes include moving objects having relatively large motions with respect to the background, the object motions should be rejected as they do not really reflect global motion of the camera. Since we are not interested in extracting the exact shape of the object, sparse motion vectors are sufficient to get the motion characteristics of the video object. Working with sparse motion vectors greatly reduces the computational burden. Because we will further adopt robust motion estimation methods later in the global motion estimation, here we adopt the following coarse object segmentation to reduce processing times.

First, compute the derivative of motion vectors. If the derivative is greater than some predefined threshold, it may be object boundary. Therefore the block position is marked as a candidate. In the next step, we employ area propagation based on these candidate points. Blocks with similar amplitude and phase angle are combined:

$$| \arg(D(x,y)) - \arg(D_k) | < T_{angle} ,$$
$$| \, \| D(x,y) \| - \| D_k \| \, | < T_{norm} , \quad (8)$$

where $\arg(\cdot)$ and $\| \cdot \|$ denote angle and amplitude of a vector. T_{angle} and T_{norm} are predefined thresholds. $D(x, y)$ are motion vectors of computed blocks. D_k are mean values of propagation areas. The final objects are the combined area. If an object area is relatively small, it is considered as noise, therefore skipped in advance. By means of a similar convolution mask as previous spatial domain, we can remove motion vectors of moving objects.

Through the above spatial and temporal processing, the remaining motion vectors are mostly reliable, which are now available for estimating the camera's global motion parameters.

3.3 Global motion estimation

According to affine motion model (1), we can deduce the following relationship between motion vectors and the pixel's position:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = AX + T = \begin{pmatrix} a_1 - 1 & a_2 \\ a_3 & a_4 - 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} , \quad (9)$$

where Δx and Δy refer to the motion vector of every block's center position. As filtered motion vectors may still include outliers, we use a robust M-estimation to deal with them. In an M-estimation formulation, the unknown parameters are estimated by minimizing an objective function of the residual error. That is,

$$\min \sum_{a^p \in R} \rho(\varepsilon)^2 , \ \varepsilon = [\Delta X - AX - T]. \quad (10)$$

Here $\rho(\varepsilon)$ is a robust function such as Geman-McLure function[5]. Hence, the motion estimation task becomes a minimization problem for finding global motion parameters.

However, the M-estimator has a drawback compared to a non-robust approach (least-squares), that is, the additionally introduced computational complexity. We therefore simplify the robust estimation as follows:

$$\sum w(\varepsilon)\varepsilon^2 \rightarrow \min, \tag{11}$$

$$w(\varepsilon) = \begin{cases} 1, & \varepsilon^2 < c\mu, \\ 0, & \varepsilon^2 \geq c\mu, \end{cases} \quad \mu = \frac{1}{N}\sum \varepsilon^2, \tag{12}$$

where a constant c is used to adjust the sensitivity of the algorithm. There are many iterative descending algorithms to solve this minimization problem such as the Gauss-Newton (G-N) algorithm, SOR algorithm and L-M algorithm as used in MPEG-4 VM[4]. These algorithms first compute a descending direction, and then compute the optimal increment step along this direction using a line search method. After a number of iterations, the motion parameters will converge to a set of values.

3.4 Motion filtering

A motion filter is used to smooth the unwanted camera motions noise. In absence of knowledge of the camera's motion, we make an assumptions that intentional camera motion, such as zooming, panning, translation motion with respect to the scene, is usually smooth with slow variations from frame to frame (in the time domain). On the other hand, unwanted, parasitic camera motion involves rapid motion variations over time. That is to say, high frequency components in the motion vector variations over time are considered to be effects of unwanted camera motion. Therefore, recovery of the intentional motion parameters can be achieved by using a low-pass filter.

Based on the above analysis, we use a temporal average filter (TA) of a length N (in our case, N = 5-9) for smoothing. This method is equivalent to taking average values of global motion parameters in the time domain, which can greatly reduce noise of unwanted motion.

3.5 Motion correction

After getting the raw global motion parameters and smoothed global motion parameters, we can use Eqs. (4) and (5) to correct the raw video sequence. But there are two problems in the correction procedure: error propagation and artifacts of edge area.

Since the first frame is taken as the reference and each stabilized frame is built upon previous stabilized frame, it is likely that an error caused by the motion compensation at the beginning of the video sequence will propagate to subsequent frames. Accumulation of such errors can be quite significant. To remove these annoying artifacts, error propagation control techniques must be used.

At each frame instance, we sum the error of intensity between original sequence I_i and stabilized sequence I_i^c.

$$\sum_{j=1}^{i-1} |I_j^c - I_j| < t. \tag{13}$$

If these errors exceed a predefined threshold t, the corresponding original frame is used as a new reference to "synchronize" the video sequence instead of using the current stabilized frame to estimate next frame. A small threshold value shows there is no evident abrupt movement resulted when replacing the stabilized frame with original frame. On the other hand, by using a large threshold value, it mean that we will resynchronize video sequence frequently, therefore less error propagation is expected. The value of t is set by experiments and can change with different scene.

Such a scheme provides a tradeoff between smoothness and artifacts. We also choose to force frame synchronization every 12 frames in the case where the above scheme is not invoked just like MPEG's inter-frame encoding scheme.

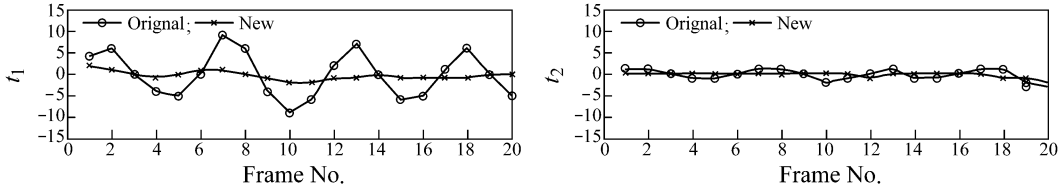
4 Experiments

To test the proposed algorithms, we use several typical video sequences, which includes clear unwanted motion. Fig. 2 shows the car sequences used in Ref. [1], where camera paned acutely around the object car. The first row of Fig. 2 (a) is the 4th, 10th, 15th, and 20th frames of the raw sequence. The second row is the corresponding corrected frames. We can find that the car is relatively still after correction (Note the relative position between the white line and the car plate). Fig. 2 (b) shows the temporal change trend of the two most important parameters: t_1 and t_2 of Eq. (1), which reflects a translation of the camera. Fig. 3 (a) shows the dynamic sequence used in Ref. [2], in which there is a moving car with camera motion including translation, rotation and scaling. The first row

of Fig. 3 (a) is the 2nd, 6th, 9th, and 12th frames of the raw sequence. The second row shows the corrected frames. As only still snapshot are shown here, difference is not obvious (but note the black area in the new images). Actual videos show that the method can efficiently reduce unwanted noise. Fig. 2 (b) shows a temporal change trend of all six parameters.

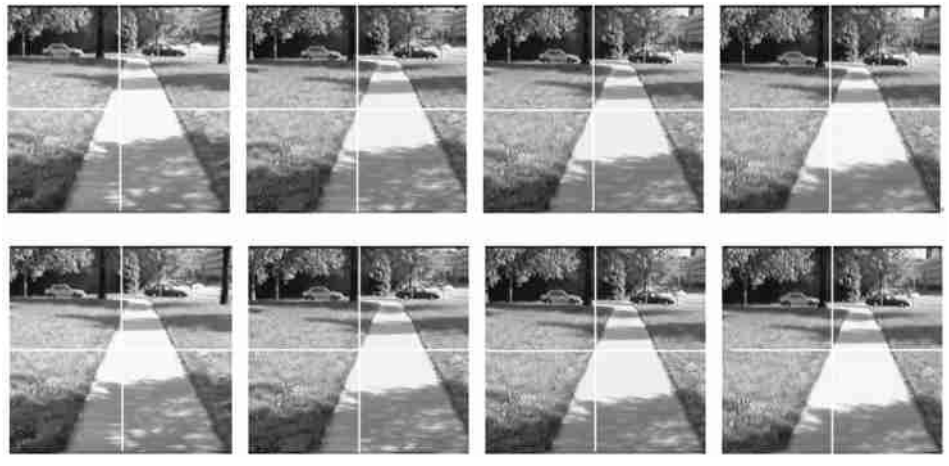
Evaluation and comparison of video stabilization algorithms is a difficult task as true motion data is difficult to obtain for real sequences. Perceptual judgment of stabilization is the best option to evaluate video stabilization algorithms aimed at human observer. Perceptual experiments show that the proposed method can be implemented in real-time with high quality.

(a) The fist row is the 4th, 10th, 15th, and 20th frames of the raw sequence and the second row shows the corresponding corrected frames

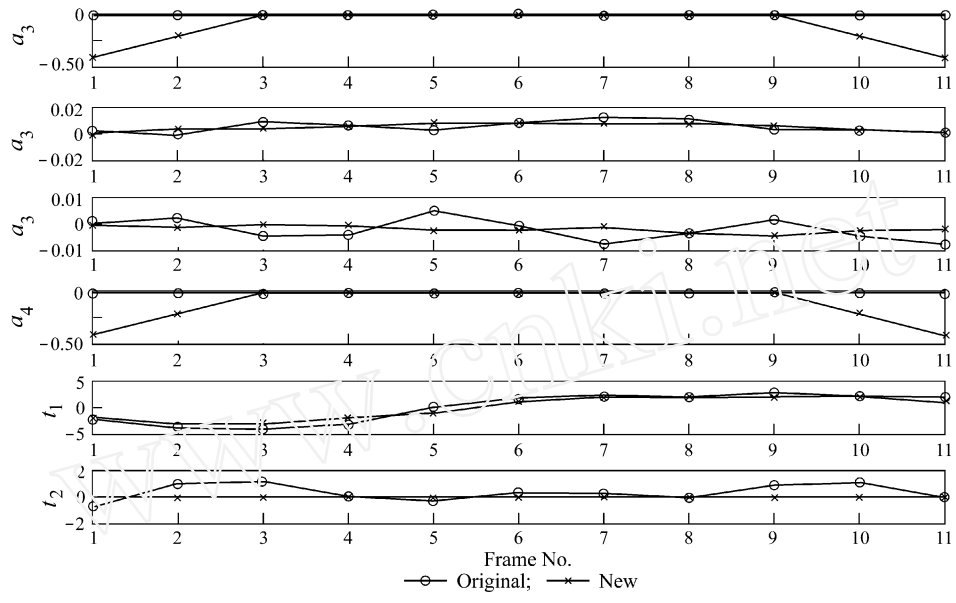


(b) Trend of temporal change in partial global motion parameters

Fig. 2 Car test sequence



(a) The fist row is the 2nd, 6th, 9th, and 12th frames of the raw sequence and the second row gives the corresponding corrected frames



(b) Trend of temporal changes in all global motion parameters

Fig. 3 Dynamic scene test sequences

5 Conclusions

On the basis of a video stabilization theory, this paper proposes an effective approach to remove unwanted motion form raw sequences. The proposed algorithm first computes semi-dense motion vectors using a blocked-based motion estimation. In order to reduce errors, we employ spatial-temporal filters to remove unreliable motion vectors. Afterwards, we compute global motion parameters of successive frames, and then adopt a temporal average filter to get smooth parameters. In the final correction processing, we use raw global motion parameters and smooth motion parameters to get stable video sequences. The proposed algorithm can be extended in several aspects: first, the global motion parameters are based on raw motion vectors, so one can use the approach to the compression domain, for example to directly process encoded MPEG streams; secondly, one can further segment moving objects from sequences and doing sprite coding.

References

[1] Censi A, Fusiello A, Roberto V. Image stabilization by features tracking image analysis and processing[A]. International Conference on Image Analysis and Processing (ICIAP '99)[C]. Sep. 1999, 665 - 667.

[2] Srinivasan S, Chellappa R. Image stabilization and mosaicing using the overlapped basis optical flow field Image processing[A]. Proc. ICIP '1997[C]. Oct. 1997, **3**(26 - 29): 356 - 359.

[3] Smolic A, Sikora T, Ohm J R. Long-term global motion estimation and its application for sprite coding, content description, and segmentation[J]. IEEE Transaction of Circuit and System for Video Technology, Dec. 1999, **9**: 1227 - 1242.

[4] MPEG-4 Video Group. MPEG-4 Video Verification Model Version 16. 0 [P]. ISO/IECJTC1/SC29/WG11, MPEG2000/N3312, Noordwijkerhout, Netherlands, 2000, 357 - 360.

[5] Sahney H S, Ayer S, Gorkani M. Model-based 2D &3D domain motion estimation for mosaicing and video representation [A]. Proc. ICCV '95 [C]. 1995, 583 - 590.

[6] Jones R C, DeMenthon D, Doermann D S. Building mosaic from video using MPEG motion vectors[A]. Proceeding of the 7th ACM Internationed Conferce on Multimedia (part 2)[C]. 1999, 29 - 32.

(Editor YAO Yue-yuan)