15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, September 3-7, 2007, copyright by EURASIP

ROBUST METHOD OF VIDEO STABILIZATION

Marius Tico and Markku Vehvilainen

Nokia Research Center
P.O.Box 100, FI-33721 Tampere, Finland
Email: tico@ieee.org

ABSTRACT

Video stabilization objective is to remove the unwanted motion fluctuations from video data. Typically, this is achieved by applying a certain amount of corrective motion displacement onto each video frame, such that to cancel the effect of high frequency fluctuations caused by unwanted camera motions. In this paper we present a robust video stabilization system comprising three operations: motion estimation, motion filtering, and motion correction. A motion estimator robust to moving objects in the scene is designed in order to identify the camera motion component that follows to be stabilized by the system. A critical role in the quality of the stabilization is then played by the ability of the system to correctly distinguish between the unwanted and the intended motion of the camera. To do this we develop a procedure that extends the Kalman filtering algorithm by incorporating the practical system constraints with respect to the amount of the corrective motion that can be applied on each video frame. The experimental results show the ability of the proposed algorithm to reduce the unwanted motion fluctuations and, at the same time, to follow the user intentional motion. Robustness with respect to large moving objects in the scene, as well as the ability of the proposed method to stabilize in the presence of motion constraints are demonstrated through a series of experiments and comparisons.

1. INTRODUCTION

The ongoing development and miniaturization of consumer devices that have video acquisition capabilities increases the need for robust and efficient video stabilization solutions. Video stabilization operation aims to remove the effect of unwanted motion fluctuations from the video data. In the context of hand held video cameras such unwanted motion fluctuations are typically caused by undesired shakes of the hand during video capturing.

A video stabilization system comprises three components: motion estimation, motion filtering, and motion correction. The estimation of the global camera motion can be accomplished either by using hardware motion sensors [1], or by employing a software approach. In the second case, one can chose a certain parametric model for the global motion, following then to estimate the model parameters by comparing consecutive frames of the video sequence. The motion between video frames could be described by various models, e.g. a translation, a rigid transformation comprising translation and rotation, or an affine transformation [2]. The translational model has been proven to be quite effective for the purpose of video stabilization operation [1, 3, 4], such that, given also its low computational complexity, it is often preferred in the consumer devices.

The motion estimator must identify the motion caused by the camera movement, called "camera motion", which must be then stabilized by the system. This problem is rather trivial when using hardware motion sensors. However, a pure software solution, that estimates the motion by matching the visual information in different video frames, could be disturbed by large moving objects passing in front of the camera. In order to identify the camera motion, the algorithm should distinguish between the part of the scene that is expected to be static with respect to the camera (e.g. background), and other parts of the scene representing various moving objects. In static scenes the problem is trivial as long as the entire scene is static with respect to the camera. However, video materials are often capturing dynamic scenes, in which the identification of camera motion is quite challenging due to moving objects passing in front of the camera.

Once detected, the camera motion comprises two components: the user intended motion, and the unwanted motion. The objective of motion filtering operation is to distinguish between these two motion components such that to allow subsequent compensation only for the undesired motion. For this, it is typically assumed that the intended motion component is smooth, such that it can be calculated by low-pass filtering the estimated global motion.

In [5] the authors proposed low-pass filtering the camera motion trajectory in Fourier domain. The solution provides a smooth stabilized motion and it can be applied for off-line stabilization of a recorded video sequence. Unfortunately, the solution is unsuitable for a real-time implementation on a typical consumer device due to its large memory requirements needed to store several frames of the input video sequence. For a real-time implementation a causal low-pass filter is preferred in order to reduce the memory requirements to a minimum.

First order IIR (infinite impulse response) low pass filtering system, known as Motion Vector Integration (MVI), is used in [3]. The main drawback of MVI consists of its tradeoff between smoothness of the resulted stabilized motion and the delay in reaction with respect to changes in the intended motion. The damping coefficient of the filter must be selected such that to cope with this tradeoff. Second order IIR filter, inertial filter, has been proposed in [6], as an attempt to reduce the phase delay, and hence to enhance the ability to follow any intended changes in the camera motion. Kalman filtering procedure has been used for video stabilization in [4, 7], and it has been proven to be a simple and robust solution for on-line video stabilization implementations.

The final stage of the video stabilization system, namely motion correction, consists of geometrically transforming each video frame such that to cancel the effect of unwanted motion. In particular, using a translational motion model, the motion correction is accomplished by correcting the position

of the video frame with an amount equal with the difference between the smoothed translational motion parameters and the estimated global motion parameters. However shifting the position of the input frame may determine some regions of the output stabilized frame to be undefined. In order to prevent this, a number of additional border pixels can be allocated in the input frame, following then to create the output frame by cropping it from the larger input frame. The amount of corrective displacements that can be applied on each video frame is thereby limited in practice. Any larger corrective displacement would result into an incomplete output image, as long as part of this image falls outside the boundaries of the input frame. On the other hand, if the amount of corrective displacement is truncated to the maximum value allowed by the system then the output frame may not be correctly stabilized.

In this paper we present a video stabilization system, emphasizing the design of the motion estimation and motion filtering operations that play a critical role in the stabilization quality. The proposed methods are design to achieve robustness with respect to dynamic scenes as well as with respect to the motion constraints imposed in a practical implementation.

2. THE PROPOSED VIDEO STABILIZATION ALGORITHM

The general diagram of our video stabilization system is shown in Fig. 1. The processing could be designed to take place either off-line or on-line. The former alternative is referring to the case when the entire original (non-stabilized) video sequence is available and can be scanned several times in any order. The second alternative, namely on-line implementation, is more practical as it does not require the separate storage of the original video. The processing is carried out during the time the video data is acquired following to encode and store only the stabilized video stream.

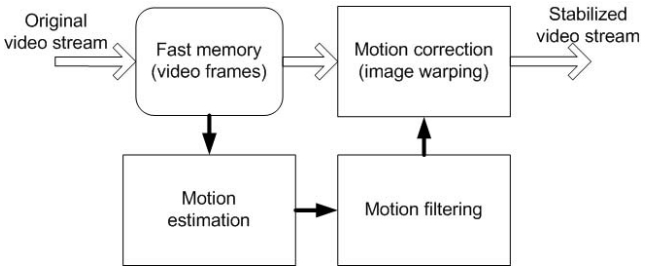

Figure 1: The diagram of the video stabilization system.

2.1 Camera motion estimation

The camera motion estimation is essentially an image registration problem, where the images to be registered are successive frames in the video stream. In our system we adopted the translational motion model that proves quite effective and suitable for mobile implementations.

Given two consecutive video frames R and I the global motion parameters that overlap I over R are estimated based on feature points representing prominent image features. The feature points are localized in the reference image frame (R), as the centers of image blocks of R, that contain prominent edges or corners. To do this, the reference image R is first divided in non-overlapping blocks (e.g. of size 16×16 pixels), and the following approximation of the Hessian matrix is calculated in each block:

$$\mathbf{W}_n = \begin{bmatrix} \overline{R_x(\mathbf{x}_n) \cdot R_x(\mathbf{x}_n)} & \overline{R_x(\mathbf{x}_n) \cdot R_y(\mathbf{x}_n)} \\ \overline{R_x(\mathbf{x}_n) \cdot R_y(\mathbf{x}_n)} & \overline{R_y(\mathbf{x}_n) \cdot R_y(\mathbf{x}_n)} \end{bmatrix}, \quad (1)$$

where $\mathbf{x}_n = (x_n, y_n)^T$ denotes the n-th block center coordinates, R_x, R_y denote the image derivatives along the horizontal and vertical direction respectively, and the over-line notations denote the average of the respective values inside the corresponding image block.

A measure of the block distinctiveness, and hence of its significance as a potential feature, is the trace of the matrix $\mathbf{W}$, which is larger in blocks that contain corners, or significant edges. The centers of the blocks for which trace($\mathbf{W}$) exceeds a given threshold are thereby selected in our implementation as features points to be used in the registration process.

Given an input image frame (I), a block motion vector $\mathbf{d}_n = (\Delta x_n, \Delta y_n)^T$ is calculated for each feature point $\mathbf{x}_n$ of R. To do this, we employ a block matching procedure [2], that identifies the most similar block in I with the given reference block. After this procedure we may end up with a number of motion vectors that are not valid for estimating the global motion between the two frames. Thus, some motion vectors may originate (or end) in objects that appear in only one of the two images, as it is the case with fast moving objects in the scene. Also other motion vectors may be erroneously estimated due to various image degradations (e.g. noise and motion blur). In order to identify and select the correct motion vectors we employ the RANSAC (Random Sample Consensus) algorithm [8], in correlation with our assumption that the global motion between the two frames follows a translational motion model. Given the selected motion vectors, the weighted Least Squares estimate of the parameters of the global motion between the two image frames is given by

$$\Theta = \left[\sum_n \mathbf{W}_n\right]^{-1} \left[\sum_n \mathbf{W}_n \mathbf{d}_n\right], \quad (2)$$

where Θ denotes the estimated translational vector, and the sums are only over the valid motion vectors identify by the RANSAC procedure.

2.2 Motion filtering

Let z_n denotes one of the motion parameters estimated based on the n-th frame of the video sequence. We can write that

$$z_n = s_n + u_n, \quad (3)$$

where s_n and u_n stand respectively for the intended and unwanted components of the motion parameter. Thus, at moment n, a correction of $c_n = -u_n$ is needed in order to stabilize the current frame. In a practical implementation the amount of corrective motion is limited, i.e. $|c_n| \leq d$, where d represents the number of additional border pixels along the given direction.

A state space representation model for the motion parameter can be assumed as follows

$$\begin{aligned} \mathbf{x}_n &= \mathbf{A}\mathbf{x}_{n-1} + \mathbf{b}e_n, \\ z_n &= \mathbf{c}^T \mathbf{x}_n + u_n, \end{aligned} \quad (4)$$

where e_n and u_n are the process and measurement noise terms that are assumed zero mean Gaussian distributed with variances σ_e^2 and σ_u^2 respectively. The process matrix $\mathbf{A}$ has size $K \times K$, and the vectors $\mathbf{c}$ and $\mathbf{b}$ are of size $K \times 1$. The state $\mathbf{x}_n$ is a $K \times 1$ vector from which the intended motion s_n can be extracted by $s_n = \mathbf{c}^T \mathbf{x}_n$.

Kalman filtering procedure provides the optimal estimate $\hat{\mathbf{x}}_n$, (and ultimately the optimal estimate of s_n), based on the assumed model (4). In our work we extend the procedure by incorporating the constraint $|c_n| \leq d$. Thus, the following algorithm is call at each iteration in order to calculate the current motion correction based on estimated frame position z_n.

---

1. $\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{g}\left(z_n - \mathbf{c}^T \mathbf{A}\mathbf{x}_{n-1}\right)$
2. $u_n = z_n - \mathbf{c}^T \mathbf{x}_n$
3. if $|u_n| > d$ then
4. $\quad \mathbf{x}_n = \mathbf{x}_n + \text{sign}(u_n)(|u_n| - d)\mathbf{P}\mathbf{c}(\mathbf{c}^T\mathbf{P}\mathbf{c})^{-1}$
5. $\quad u_n = z_n - \mathbf{c}^T \mathbf{x}_n$
6. return $-\, u_n$

The Kalman gain matrix $\mathbf{g}$ and the matrix $\mathbf{P}$ used in the algorithm are independent of the input data. They are precomputed by iterating the following equations until convergence,

$$\begin{aligned} \hat{\mathbf{P}} &= \mathbf{A}\mathbf{P}\mathbf{A}^T + \sigma_e^2 \mathbf{b}\mathbf{b}^T \\ \mathbf{g} &= \hat{\mathbf{P}}\mathbf{c}\left(\mathbf{c}^T\hat{\mathbf{P}}\mathbf{c} + \sigma_u^2\right)^{-1} \\ \mathbf{P} &= \left(\mathbf{I}_K - \mathbf{g}\mathbf{c}^T\right)\hat{\mathbf{P}} \end{aligned} \tag{5}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix, and initial value of P could be the identity matrix.

In contrast to our solution, a trivial approach would consists of running Kalman filtering procedure, following to disregard the resulted correction whenever it exceeds the system constraint. In such cases, the correction recommended by the filtering procedure is simply truncated to the maximum corrective value allowed by the system.

In our experiments we employed the state space model that assumes a constant velocity camera motion along each direction between moments of user intentional motion changes [4]. The model parameters are:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \ \mathbf{c} = [1\ 0]^T, \text{ and } \mathbf{b} = [1\ 1]^T. \tag{6}$$

3. EXPERIMENTS

We evaluated the proposed algorithm on several video sequences taken in various usage conditions, e.g. standing, walking, running, etc. A typical result is shown in Fig. 2, where the motion trajectory stabilized by the proposed algorithm exhibits less jitter than the original motion, and at the same time, it follows closely the most probable user intended motion. Both qualities are essential for the acceptability in consumer devices. For instance the video sequence whose motion is shown in Fig. 2 exhibits a significant panning motion (horizontal motion) that should be followed by the system.

Maintaining the stabilized motion trajectory as close to the original motion as possible has also the advantage to reduce the amount of corrective motion needed for aligning

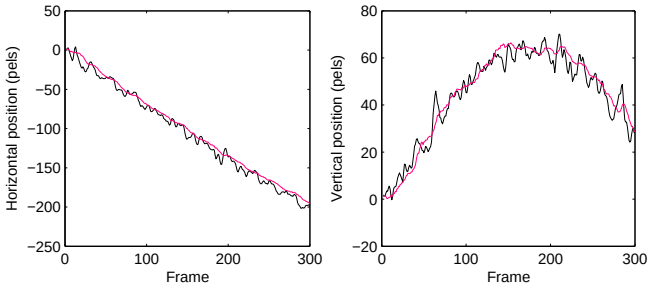

Figure 2: Example of stabilized motion trajectory: original motion trajectory (black), and the stabilized motion trajectory delivered by the proposed algorithm (red).

each frame. However, in extreme cases of fast motion, the maximum corrective displacement allowed by the system can be reached. In such a case the proposed constraint equation embedded in the filtering procedure is executed. In order to evaluate the effectiveness of our method for constraint treatment we stabilized a video sequence under different constraint sizes (i.e. different numbers of border pixels). The stabilization quality in each case is estimated by comparing the amount of "jitter" present in the motion trajectory before and after stabilization. To do this, we estimate the jitter energy present in a motion trajectory as the variance of a high pass filtered version of that trajectory. Next, as an objective measure of the stabilization quality we use "jitter attenuation", that is defined as the ratio between the jitter energy in the original and stabilized motion trajectory.

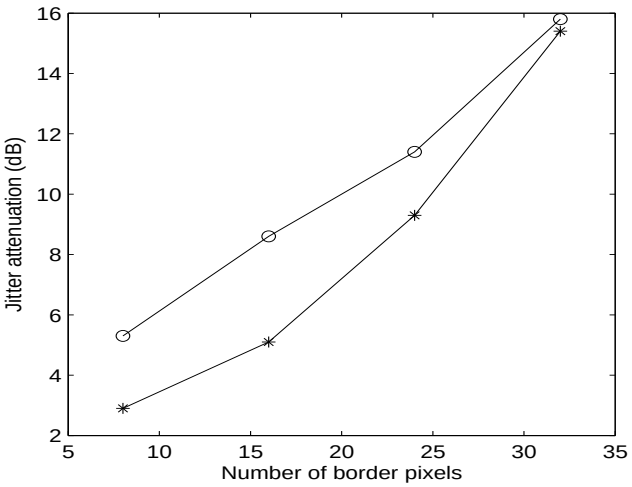

Figure 3: Jitter attenuation under different system constraints when using: the proposed method for constraint integration (circle mark), and the saturation of the motion correction at the maximum value allowed by the system (star mark).

Fig. 3 shows the jitter attenuation achieved for different values of the constraint (number of border pixels), when using either the proposed solution, or a trivial saturation of the motion correction to the maximum value allowed in the system. All simulations have been performed using the same parameters for the state space model, such that the differences in performance are caused only by the strategy used to incorporate the system constraint into the filtering proce-

(a) (b)

Figure 4: Several overlapped video frames after stabilization: (a) when using our method, (b) when using another method proposed in the literature. Note that the proposed approach is not disturbed by the moving object, being able to stabilize with respect to the static background that is sharp in the overlapped picture.

dure. The simulation shows that in the presence of system constraints the stabilization quality is improved by employing the proposed solution. We note also that by increasing the number of border pixels the two solutions converged to similar performance as the constraint is less challenged. However increasing the number of border pixel is not always possible since it determines also an increase in the computational load of the imaging system that has to provide to the stabilizer a larger frame size at the same frame rate.

The robustness of our algorithm with respect to large moving objects in the scene is illustrated in Fig. 4. In the presence of moving objects the algorithm is expected to stabilize with respect to the static background, and it should not attempt to ”follow” the moving object that is passing in front of the camera. Overlapping the stabilized frames of a short video sequence, we note that our algorithm is able to stabilize with respect to the background, which is sharp in the overlapped image, shown in Fig. 4 (a). For comparison, Fig. 4 (d) shows the overlapped frames stabilized with the registration algorithm in [9]. We observe that, in this case, the overlap image (Fig. 4 (d)) is blurred everywhere as the algorithm is more disturbed by the moving object.

4. CONCLUSIONS

In this paper we presented a video stabilization system emphasizing the design of motion estimation, and motion filtering components. Aiming for a robust video stabilization solution we designed a global motion estimator that is able to distinguish the motion caused by the camera movement from the motion of any possible moving object that may appear in the scene. In order to identify the unwanted motion component we proposed a filtering algorithm that takes into consideration the system constraints with respect to the amount of the corrective motion that can be applied on each frame. An efficient implementation of the motion filter is achieved by pre-calculating the steady-state filter parameters. The proposed stabilization algorithm has been demonstrated through a series of experiments and comparisons carried out on real video data.

REFERENCES

[1] M. Oshima, T. Hayashi, S. Fujioka, T. Inaji, M. Mitani, J. Kajino, K. Ikeda, and K. Komoda, “VHS camcorder with electronic image stabilizer,” *IEEE Transaction on Consumer Electronics*, vol. 35, no. 4, pp. 749–758, 1989.

[2] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Prentice Hall, 2002.

[3] S. J. Ko, S. H. Lee, and K. H. Lee, “Digital image stabilizing algorithms based on bit-plane matching,” *IEEE Transaction on Consumer Electronics*, vol. 44, no. 3, pp. 617–622, 1998.

[4] S. Erturk, “Digital image stabilization with sub-image phase correlation based global motion estimation,” *IEEE Transaction on Consumer Electronics*, vol. 49, no. 4, pp. 1320–1325, 2003.

[5] S. Erturk and T. Dennis, “Image sequence stabilization based on DFT filtering,” *IEE Proc. On Vision Image and Signal Processing*, vol. 147, no. 2, pp. 95–102, 2000.

[6] J. S. Jin, Z. Zhu, and G. Xu, “A Stable Vision System for Moving Vehicles,” *IEEE Transaction on Intelligent Transportation Systems*, vol. 1, no. 1, pp. 32–39, 2000.

[7] A. Litvin, J. Konrad, and W. C. Karl, “Probabilistic video stabilization using Kalman filtering and mosaicking,” in *Proc. of SPIE Electronic Imaging*, vol. 5022, 2003, pp. 663–674.

[8] M. A. Fischler and R. C. Bolles, “A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Comm. of the ACM*, vol. 24, pp. 381–395, 1981.

[9] S. Erturk, “Translation, rotation and scale stabilisation of image sequences,” *Electronics Letters*, vol. 39, no. 17, pp. 1245–1246, 2003.