

Integration of Digital Stabilizer With Video Codec for Digital Video Cameras

Homer H. Chen, *Fellow, IEEE*, Chia-Kai Liang, *Student Member, IEEE*, Yu-Chun Peng, and Hung-An Chang

Abstract—This paper presents three novel schemes for integrating digital stabilizer with video codec of a digital video camera, each designed for a different application scenario. Scheme 1 confines the global motion estimation (ME) to within a small background region determined by clustering the motion vectors (MVs) generated by the video encoder. Scheme 2 performs the main ME task at the digital stabilizer and sends the resulting motion vectors to the video encoder, where the motion vectors are further refined to subpixel accuracy. Scheme 3, which is applied on the decoder side, acquires the motion information directly from the video decoder to perform digital stabilization. These integration schemes achieve computational efficiency without affecting the performance of video coding and digital stabilization.

Index Terms—Digital video camera, digital video stabilization, global motion estimation (ME), ME, video coding.

I. INTRODUCTION

THE MOVEMENT of a video camera caused by hand shaking or platform movement often introduces jerky image motion that is annoying to human eyes. The undesirable effect of hand shake is even more profound during zooming. The purpose of stabilization is to remove such unwanted motion from the image sequence while preserving the dominant motions such as intentional camera panning.

Solutions to the stabilization problem involve either hardware or software to compensate for the unwanted camera motion. In early years, mechanical image stabilization devices such as accelerators, gyro-sensors, and dampers [1] were used. Optical image stabilizers, which have been developed recently, employ an optical device such as a prism or a movable lens assembly that moves opposite the camera movement [2]. Like the mechanical stabilizers, the optical image stabilizers are generally expensive and lack the kind of compactness that is crucial for today's consumer electronic devices. On the contrary, digital stabilizers use video signal processing techniques to solve the

problem electronically without resorting to mechanical or optical devices. This type of approaches is more economic for digital video cameras.

Both the video encoder and the digital stabilizer of a digital video camera need to compute the image motion. Motion estimation (ME) can take up 90% of the total computation of a digital stabilizer and 50%–70% of a video codec. Combined together, the operation required for ME can consume more than 70% of the total computation.

In the context of image stabilization, most previous approaches attempt to reduce the computational cost of ME by using fast algorithms or by limiting the global ME to small, pre-defined regions [3], [7]–[9], [15], [20]. Such approaches consider digital image stabilization and video encoding separately and attempt to trade the accuracy of motion vectors (MVs) for the computational efficiency.

We propose to integrate digital stabilizer with video codec by making the two modules of a digital video camera share a common ME process. This work is motivated by a need for efficient algorithms for system-on-chip (SoC) implementation of the digital video camera. Three integration schemes are presented in this paper, each for a different application scenario. These integration schemes are built upon a simple but effective digital stabilization algorithm that is comparable to gyro-based motion sensors in performance.

This paper is organized as follows. The architecture of the digital stabilizer is described in Section II. The effect of digital stabilization on the performance of video encoders is investigated in Section III. The three integration schemes proposed are described in Section IV, and the experimental results and performance evaluation of the integration schemes are given in Section V. In Section VI, the limitations and other related issues of digital stabilization are discussed. Finally, a summary is given in Section VII.

II. DIGITAL STABILIZATION

We begin this section by giving an overview of the existing digital stabilization techniques, followed by a description of our digital stabilization algorithm.

A. Previous Work

The basic structure of a digital video camera, as shown in Fig. 1, consists of an image signal processing (ISP) module, an imaging sensor, a video codec module, and a memory unit, all connected to a data bus. The image sequence captured by the camera sensor is processed frame by frame by the ISP module to handle color interpolation, color correction, white balance, image enhancement, etc. The digital stabilizer is normally placed in the ISP module. It applies a global ME to detect

Manuscript received July 17, 2006; revised November 15, 2006. This work was supported in part by the National Security Council under Grants NSC 94-2213-E-002-070 and NSC 94-2752-E-002-006-PAE. The work of C. K. Liang was supported in part by the MiTac Technology Corporation. This paper was recommended by Associate Editor C. N. Taylor.

H. H. Chen and C.-K. Liang are with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: homer@cc.ee.ntu.edu.tw).

Y.-C. Peng was with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. He is now with Novatek Inc., Hsinchu 300, Taiwan, R.O.C.

H.-A. Chang is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2007.897113

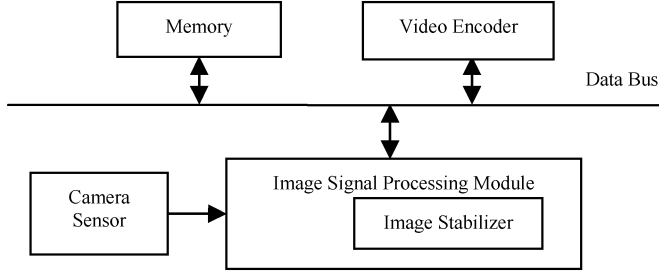


Fig. 1. Basic structure of a digital video camera.

the image motion induced by the unwanted camera movement and then compensate the induced image motion to stabilize the image sequence. Each stabilized image is then input to the video encoder.

Uomori *et al.* [3] proposed a digital stabilization system that estimates the global image motion by correlating block-based local MVs. Paik *et al.* [4], [5] proposed the estimation of global image motion based on the isolativity and stability of the local MVs determined by edge pattern matching. In [6], Egusa *et al.* developed a fuzzy theorem to control the stabilization system. In [7], a bit plane matching algorithm was applied to reduce the computational complexity of the ME while preserving its accuracy. Along the same direction, a Gray-coded bit plane matching algorithm and hardware architecture was proposed in [8] and [9] to make bit plane matching more reliable. In [10], a multiple weighted bit plane matching method was used in image stabilization applications. A lowpass filter was adopted by Ertürk to smooth the interframe global motion [11], and improved variants of the filter were reported in [12]–[14]. An inverse triangle method was proposed in [15] based on a point matching algorithm [3], whereas a wavelet transform algorithm was used in [16] to improve the performance of the digital stabilizer. A performance evaluation and characterization of various image stabilization algorithms were presented in [17] and [18], and a comparative review of digital stabilization for mobile communications was presented in [19].

B. The Proposed Algorithm

The motion field of an image sequence is contributed by three kinds of motions: 1) object motion; 2) intentional camera motion such as panning; and 3) unwanted camera motion such as hand jiggles. The main challenge of a digital stabilizer is to distinguish the jerky image motion caused by the unwanted camera motion from the object motion and from the intentional camera motion.

In our approach, the camera motion (either intentional or unwanted) is separated from the object motion first. Then the motion induced by hand shaking is removed from the resulting camera motion [33]–[36].

When the moving objects do not dominate (in terms of image size) the scene, the global image motion induced by the camera can be extracted from the motion of background pixels. Furthermore, since the unstable camera movement caused by hand jiggles usually is of higher frequencies than the intentional camera movement, smoothing the global motion by a lowpass filter generates the desired stabilization effect. When the moving objects

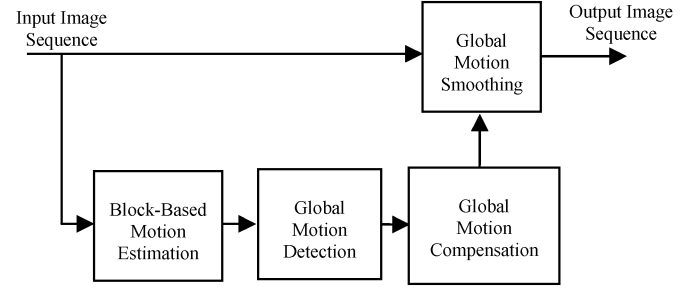


Fig. 2. Block diagram of the digital stabilizer.

dominate the scene, the global motion thus obtained is a summation of object motion and camera motion. In this case, a lowpass filtering on the global motion would also smooth the movement of the objects and improve the visual quality of the sequence.

Fig. 2 shows the architecture of the proposed digital stabilizer. First, the motion field between the current frame and the preceding frame is computed using block-based ME (BME). The resulting motion field is input to the global motion (GM) detection module, which determines the global MV (GMV) for each video frame. Then the GM is filtered by a motion smoother to remove the unstable camera movement. Finally, the current frame is motion compensated to achieve stabilization. The stabilized video frame is then input to the video encoder. Each step of the proposed digital stabilizer is described in detail next.

1) *Block-Based Motion Estimation:* Various techniques, such as edge-based, optical flow, feature-based, and phase correlation, have been proposed for GM estimation [4], [5], [20]–[23]. To be compatible with video coding standards such as MPEG-4, we adopt the BME method to determine the motion field.

Our algorithm works as follows. First, it divides the current frame into a number of MBs and computes the MV for each MB. The resulting MV points to the most correlated reference block in the previous frame within the search range. The sum of absolute difference (SAD) is used to measure the goodness of each candidate MV (x, y)

$$\text{SAD}[x, y] = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C[i, j] - S[i + x, j + y]| \quad (1)$$

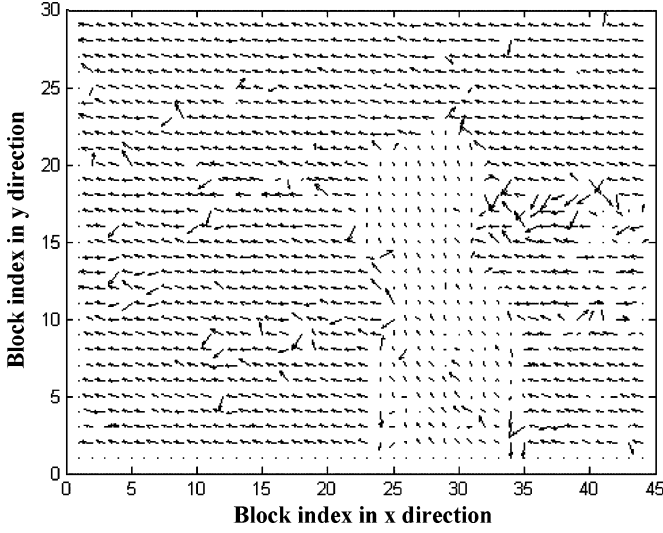
where $C[\cdot, \cdot]$ denotes the current MB, $S[\cdot, \cdot]$ the reference block, and N the block size. The candidate MV $\mathbf{v} = (x, y)$ with the smallest SAD is chosen as the MV of the current MB

$$\mathbf{v} = \arg \min_{(x, y)} \{\text{SAD}[x, y]\}, \quad -r \leq x, y \leq r - 1 \quad (2)$$

where r denotes the motion search range. An example of the motion field is shown in Fig. 3. The resulting (local) MVs are processed to determine the GM of each frame.

2) *Global Motion Detection:* The GM induced by camera movement is determined by a clustering process that consists of the following steps.

Step 1) Construct the histogram H of the local MVs. The value of $H[x, y]$ is incremented by one each time the local MV (x, y) is encountered.

Fig. 3. Motion field of frame 70 of the test sequence *Basketball*.

Step 2) Detect the peak of the histogram. Directly determining the peak value of $H[x, y]$ using the raw data is sensitive to noise. To solve the problem, a 5×5 lowpass filter is applied to H before the peak detection

$$H'[x, y] = \sum_{i=-2}^2 \sum_{j=-2}^2 H[x+i, y+j] \quad (3)$$

where $H'[x, y]$ denotes the filtered histogram.

Step 3) Determine the GMV. Let \mathbf{p} denote the local MV corresponding to the peak value of the filtered histogram H' . That is

$$\mathbf{p} = \arg \max_{(x,y)} \{H'[x, y]\}, \quad -r \leq x, y \leq r-1. \quad (4)$$

Then the GMV \mathbf{g} is obtained by computing the weighted average

$$\mathbf{g} = \frac{\sum_{i=-2}^2 \sum_{j=-2}^2 H[p_x+i, p_y+j] \cdot (\mathbf{p} - (i, j))}{\sum_{i=-2}^2 \sum_{j=-2}^2 H[p_x+i, p_y+j]}. \quad (5)$$

As an example, Fig. 4 shows the clusters detected from the motion field depicted in Fig. 3. In this example, the cluster located at $(-22, -4)$ receives the maximum vote, and the weighted average of this cluster yields the GMV.

3) *Global Motion Smoothing*: A lowpass filter similar to the one adopted in [10] is used for smoothing the GM. Since the frequency of the unwanted camera movement is usually much higher than that of the intentional camera motion, this lowpass filtering process effectively removes the unwanted camera movement from the GM. To enable real-time processing, a causal lowpass filter described by

$$\mathbf{g}'_n = \sum_{i=0}^m c_i \mathbf{g}_{n-i} \quad (6)$$

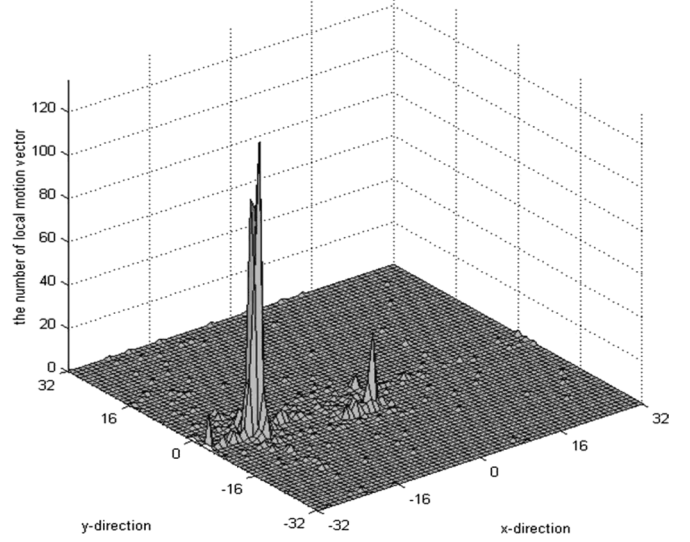


Fig. 4. Clusters of a motion field.

where \mathbf{g}_n represents the GMV of frame n , \mathbf{g}'_n the resulting smoothed motion, and c_i the normalized coefficients of the filter, is used in our algorithm.

Equation (6) can be considered as a weighted moving average filter with window size m . With this filter, \mathbf{g}'_n becomes smoother as m increases. Fig. 5(a) shows the values of \mathbf{g}_n and \mathbf{g}'_n of a test sequence in the y -direction for $m = 7$. The corresponding accumulative GM, which represents the trajectory of the camera, is shown in Fig. 5(b). The choice of m , which is empirically determined, has almost no effect on the stabilization performance. It only slightly affects the range of GM compensation.

4) *Global Motion Compensation*: Stabilization of a video frame is achieved by performing motion compensation on the video frame. Given \mathbf{g}_n and \mathbf{g}'_n of frame n , this is done by shifting the display window with respect to the previous display window by \mathbf{d}_n according to the equation

$$\mathbf{d}_n = \mathbf{g}'_n - \mathbf{g}_n. \quad (7)$$

The image area within the display window of the current video frame is cropped and output to the video encoder. In practice, the size of the display window, which is always smaller than the input video, depends on the magnitude of the unwanted camera motion. A large margin around the display window should be allocated to compensate for large hand shaking. For a fixed output video size, this effectively imposes a requirement on the resolution of the image sensor.

C. Performance Evaluation

Evaluation of the performance of a digital stabilization algorithm is a difficult task since the ground truth of hand shake is unavailable. Instead of seeking a solution to measure the hand jiggling motion, we set up a camera platform on which both our digital stabilizer and a gyro-based motion detection module¹ used in state-of-the-art digital cameras are installed.

¹Gyro-Sensor: Angular Velocity Sensor, Murata Inc. [Online]. Available: <http://www.murata.com>

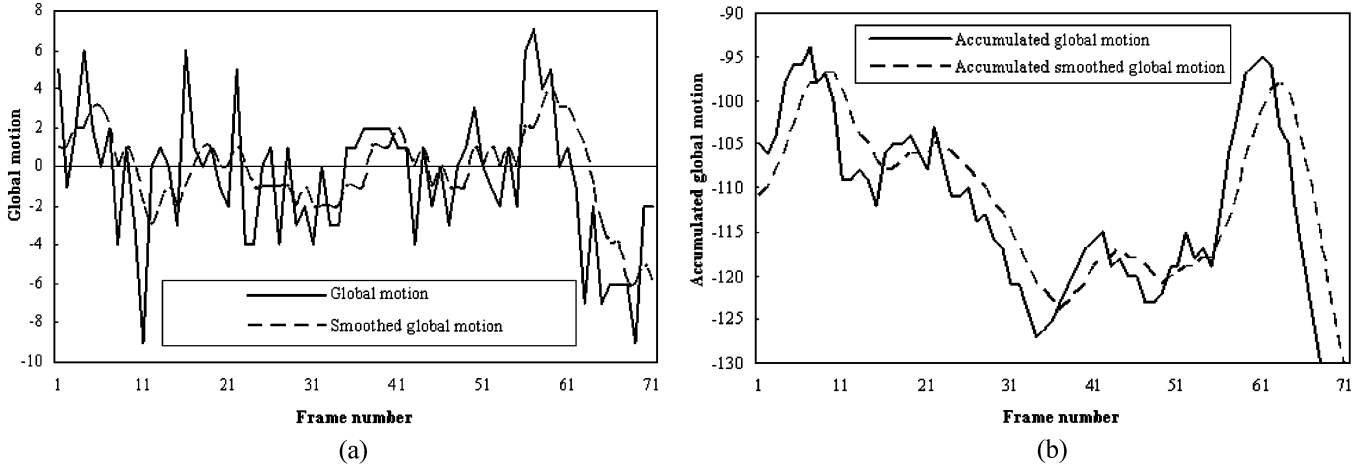


Fig. 5. (a) Global motion before and after smoothing. (b) Accumulative GM before and after smoothing. Only the y -component is depicted here.

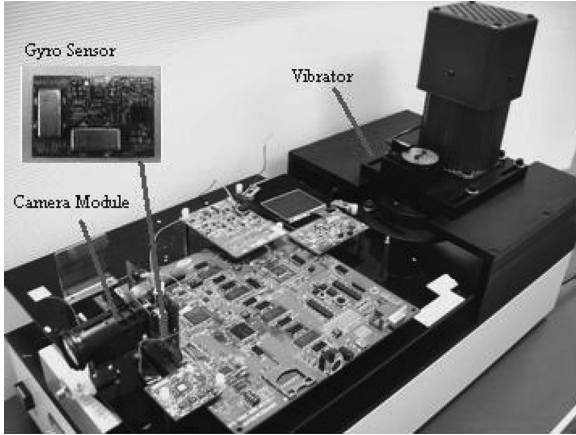


Fig. 6. Video camera testbed with gyro sensors and a vibrator.

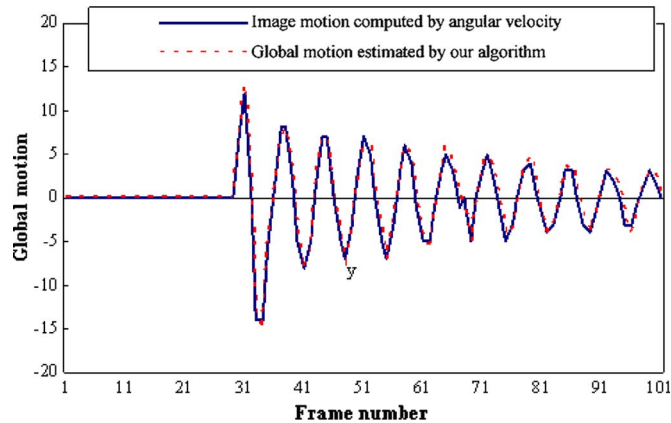


Fig. 7. Comparison of the GM obtained by our system and angular velocity sensor.

Specifically, the platform consists of a camera module to capture video, a mechanical vibrator to generate jiggling motion in the horizontal (x -) direction, and a gyro-based motion detection module mounted on the video camera to detect the camera movement introduced by the mechanical vibrator. For comparison purpose, the scene maintains still, only the camera is moving. Fig. 6 shows the camera platform designed for the

TABLE I
RMSE VALUES OF THE FOUR TEST SEQUENCES

Test Sequence	RMSE value
Office1	0.03786
Office2	0.02654
Wall	0.03324
Paper	0.04423

performance evaluation. The GM detected by our stabilization algorithm is compared with the translational MVs converted from the angular velocity generated by the gyro sensor. In our algorithm, we set the block size to 16 and the search range in each direction to 32 for ME. A window size of 9 is used for the GM smoother.

The test result is shown in Fig. 7. We can see that the performance curves of the two techniques are almost indistinguishable. The result has an important implication: digital stabilizer can reach performance comparable to that of gyro-based sensors. In our applications, digital stabilizer is attractive because of the economic reason. It should also be noted that gyro-based sensors do not respond to pure translational motions, while the digital stabilization approach does not have the problem.

To quantify the performance of our algorithm, we compute the root-mean-square error (RMSE)

$$\text{RMSE} = \frac{1}{N} \sqrt{\sum_{n=0}^N (g_n - o_n)^2} \quad (8)$$

where g_n and o_n , respectively, denote the GMs of frame n determined by our algorithm and the gyro sensor. The results are shown in Table I, which indicates that our algorithm performs consistently with an average RSME of 0.035 pixels.

Recall that the goal of image stabilization is to remove the undesired image motion caused by hand shake. Therefore, the performance of image stabilization can be judged by checking how well such image motion is removed. A stabilized image sequence should appear smoother than the raw image sequence. Consequently, the frame difference of a stabilized image sequence would be smaller than the raw image sequence, as shown in Fig. 8. We express the frame difference in terms of squared-

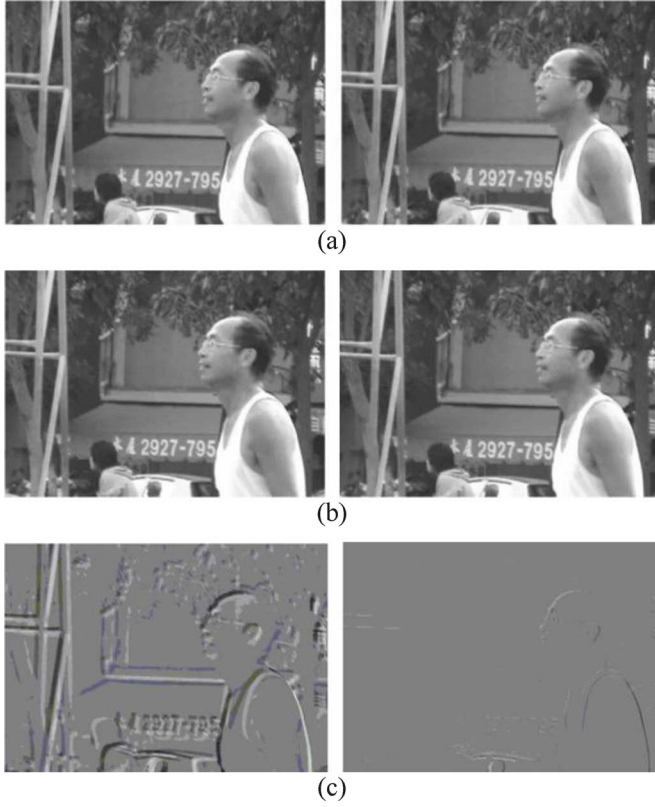


Fig. 8. Original sequence (left) and stabilized sequence (right). (a) Frame 30. (b) Frame 31. (c) Frame difference.

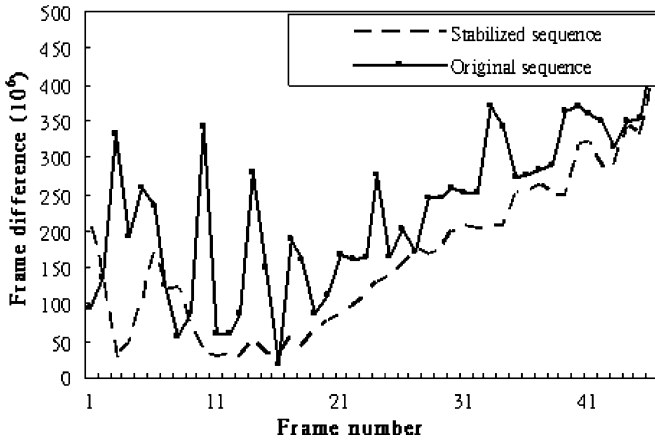


Fig. 9. Frame difference from frame 30 to frame 100 of sequence *Basketball*.

error and show the result in Fig. 9. As expected, the stabilized image sequence is smooth and does not have the jumpy behavior exhibited by the raw image sequence.

III. IMPACT OF IMAGE STABILIZATION ON VIDEO CODING

Before the digital stabilizer described in the previous section is integrated with the video codec, we need to investigate if image stabilization would impact the performance of video coding. To answer the question, we set up an experiment using two recently developed video coding standards, MPEG-4 [25] and H.264 [26], to encode the image sequences. An MPEG-4 simple profile (SP) encoder and an H.264 main profile (MP)

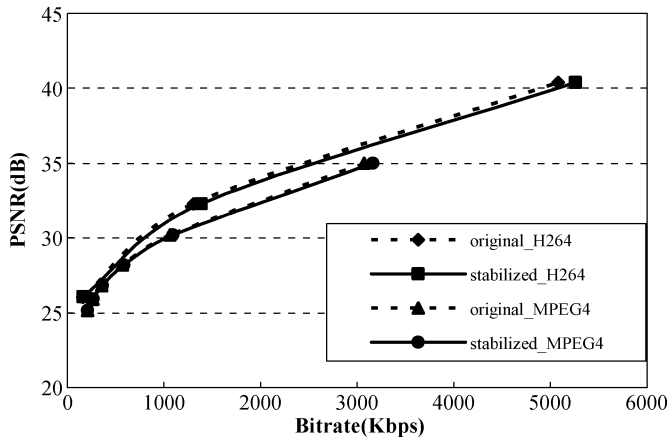
TABLE II
CODEC CONFIGURATIONS

	MPEG-4 SP	H.264 MP
Rate-control	Off	Off
QP for I frame	5	24
QP for P frame	5, 10, 15, 20, 25, 30	10, 20, 30, 40, 50
No. of reference frame(s)	1	5
Version	Microsoft	JM 8.1
Search range	[-16, +16]	
I frame period	Only one I frame	
Size of sequence <i>Game</i>	304×224	
Size of sequence <i>Lab</i>	288×208	

encoder are used. Both encoders are run with constant quantization parameters (QPs). For the MPEG-4 reference software, we turn off all advanced coding features such as bidirectional VOP and GM compensation. For the H.264 encoder, we use the JM 8.1 reference software. To have a full-scale evaluation, most coding options of H.264, such as multi-reference frames, quarter-pixel motion compensation, in-loop deblocking filter, and context-adaptive binary arithmetic coding, are all turned on in the experiment. In this way the two encoders are configured at different performance levels. Table II lists the configurations of the two encoders.

In the experiment, image sequences of various scenes captured by a hand-held camera are used as the input data. The image sequences before and after the image stabilization are encoded by the two video encoders. But before being encoded, the raw image sequences are cropped to the same size as the stabilized image sequences.

Two test sequences (300 frames each) and the coding results are shown in Figs. 10–13. Similar results are obtained for the other image sequences. Tables III and IV show the averaged PSNR and bit count. As indicated in the “Motion bit diff” column of these tables, the stabilized sequences require fewer bits for MVs. Intuitively, this makes sense since the frame-to-frame motion of the stabilized video is smoother, and hence there is less information to be coded. Image stabilization also reduces the chance of the MVs going out of the motion search range and thereby enhances the temporal smoothness and the coding efficiency of the MVs. The data in the “Bitrate diff” column of Table III and IV show that the total bit rate of the stabilized sequence is slightly higher than that of the original sequence. It should be pointed out that the test data used in our experiments are decoded sequences. Although the quality of these decompressed data is high enough for our experiments, discontinuities between MBs exist because of the quantization operation of video encoding. After stabilization, such discontinuities move off the MB boundary and require extra bits to encode. However, the increase of total bit rate is insignificant. By and large, as shown in Figs. 12 and 13, the peak signal-to-noise ratio (PSNR) difference between the stabilized video and the raw video is insignificant, indicating that image stabilization has little influence on the overall performance of video coding.

Fig. 10. Sequence *Game*.Fig. 11. Sequence *Lab*.Fig. 12. Comparison of the R-D curves of sequence *Game*.

IV. INTEGRATION OF DIGITAL STABILIZER WITH VIDEO CODEC

ME is a common operation for the stabilizer and the video encoder of a digital video camera. The purpose of the integration is to reduce the computational redundancy between the two modules as much as possible while maintaining the performance of these two modules.

Three integration scenarios are considered, each leading to a different integration scheme. In the first scenario, the digital stabilizer is modified to work with an existing video codec. In the

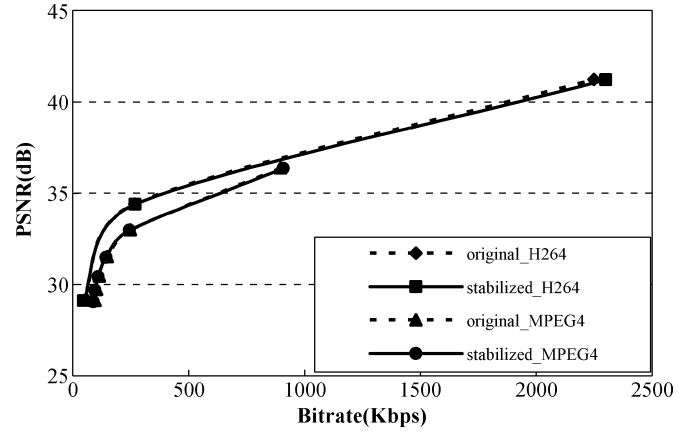
Fig. 13. Comparison of the R-D curves of sequence *Lab*.

TABLE III
PSNR AND BIT COUNTS FOR MVs OF SEQUENCE *Game* CODED BY MPEG-4

QP	Original		Stabilized		Motion bits diff	PSNR diff	Bitrate diff (%)
	PSNR	Motion bits	PSNR	Motion bits			
5	35.002	830095	35.008	823231	-6864	0.006	2.92
10	30.184	794271	30.194	782001	-12270	0.009	2.98
15	28.176	791440	28.191	771514	-19926	0.015	2.71
20	26.813	785251	26.824	765121	-20130	0.011	1.52
25	25.913	787658	25.920	765869	-21789	0.007	0.45
30	25.157	786315	25.161	766514	-19801	0.004	-0.42

TABLE IV
PSNR AND BIT COUNTS FOR MVs OF SEQUENCE *Lab* CODED BY MPEG-4

QP	Original		Stabilized		Motion bits diff	PSNR diff	Bitrate diff (%)
	PSNR	Motion bits	PSNR	Motion bits			
5	36.362	543569	36.343	517133	-26436	-0.019	0.88
10	32.993	532702	32.963	497875	-34827	-0.030	0.17
15	31.505	552357	31.474	512583	-39774	-0.032	0.22
20	30.450	560585	30.413	520769	-39816	-0.037	0.29
25	29.730	568956	29.673	528078	-40878	-0.057	0.30
30	29.106	572619	29.026	530171	-42448	-0.079	0.46

second scenario, the video encoder is modified to work with the digital stabilizer. The integer-pel MVs generated by the digital stabilizer are refined to the subpixel accuracy at the video encoder. In the third scenario, the image stabilization is performed during the decoding of a video sequence, for which the MVs are readily available from the bit stream. This scheme can be applied in the post processing stage of a video sequence captured by a digital video camera without a built-in stabilizer.

A. Scheme 1

In this scheme the digital stabilizer needs to be modified, but the video encoder remains the same. Therefore, the main task in the design of this integration scheme is to ensure that the performance of the digital stabilizer is maintained.

Delay is a major issue of this integration scheme. Since the image stabilization is performed before the video coding, the MVs received by the digital stabilizer at any time instant correspond to a previous frame. In other words, there is always a

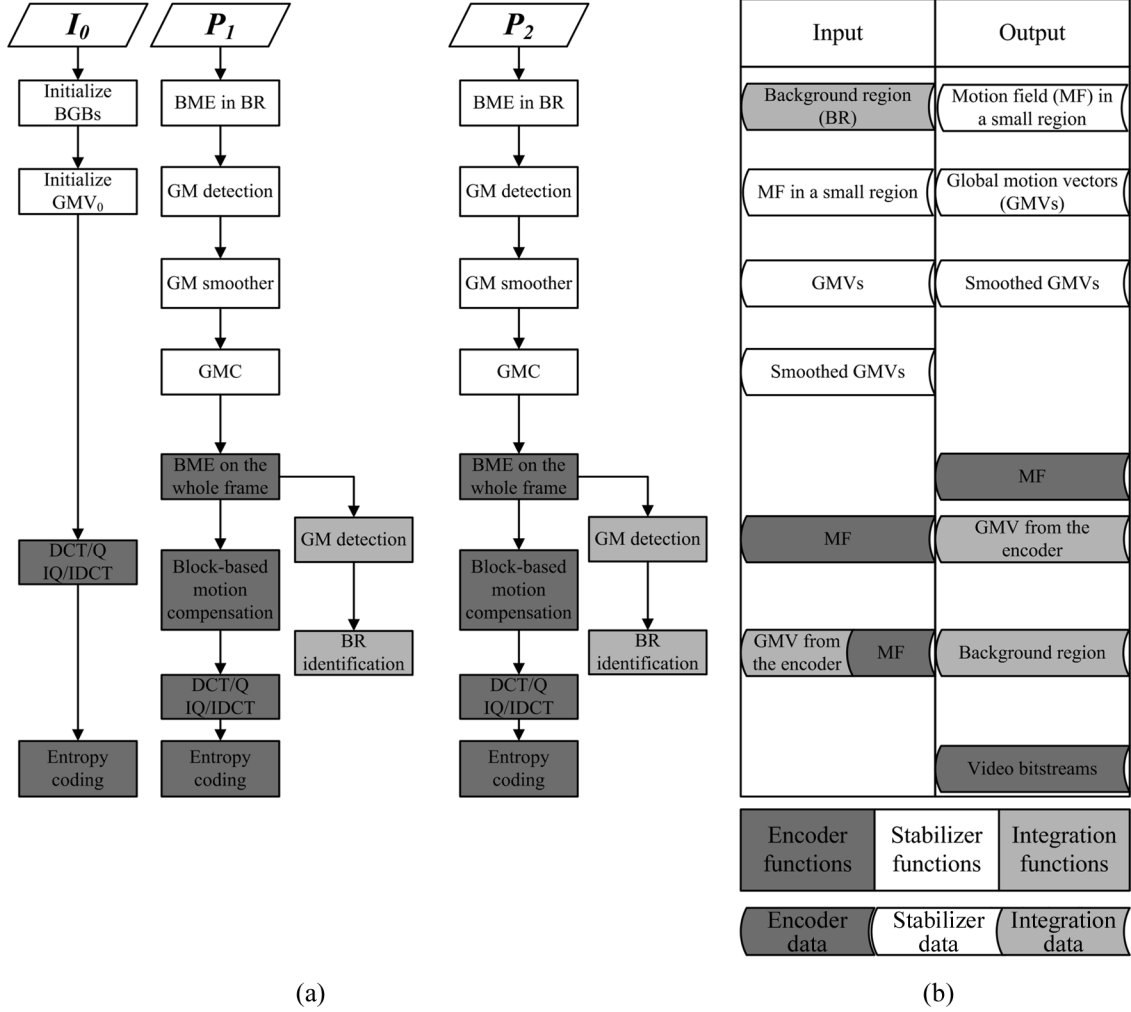


Fig. 14. System operation/data flow of Scheme 1. (a) Operation of the first three frames I_0 , P_1 , and P_2 . (b) Corresponding input and output data at each stage.

time delay. If the bi-directional prediction mode (i.e., B-frame) is turned off so that the video frames are encoded in the same order as they arrive at the video encoder, the delay is one frame time. If B-frame is enabled, the delay is more complicated. Any such delay would affect the performance of the digital stabilizer and even renders it powerless.

Therefore, the digital stabilizer has to compute the MVs of the current frame; the delayed motion data generated by the video encoder cannot be used directly for image stabilization. Obviously, the additional computation required for this step has to be justifiable, or it defeats the purpose of integration.

Most approaches to digital stabilization attempt to reduce the computational requirement of ME by confining the computation to small, predefined areas of the image. For example, the approach described in [15] assumes that background regions (BRs) are located near the boundary of the picture, and the approach described in [20] uses four small fixed regions to identify the background. Although the background of an image sequence usually does not change rapidly between neighboring frames, it cannot be guaranteed that the predefined regions always belong to the background.

We confine the ME for digital stabilization to one small region. However, instead of using a fixed region for background

prediction, we dynamically determine the BR for each video frame by clustering the MVs obtained from the video encoder. Our approach is able to adapt to the scene even if the background is moving.

The system flow of the integrated scheme is shown in Fig. 14(a), where the white blocks are the functions of the digital stabilizer and the dark gray blocks are the functions performed by the video encoder, as described in the Section II. The gray blocks in the system flow represent the new operations that are introduced as a result of the integration. The corresponding data flow of each function is shown in Fig. 14(b).

There are two new operations. The first one applies the GM detection method described in Section II-B to the MVs generated by the video encoder to determine the GM \mathbf{h} of the stabilized image. The second operation uses the resulting \mathbf{h} to determine a BR, which is basically a rectangular region consisting of one or more MBs. The BR is then used by the digital stabilizer in the next frame time to compute the GM \mathbf{g} for stabilization.

For a MB belonging to the background (or the dominant objects), its MV \mathbf{v} can be modeled as the GMV \mathbf{h} plus an additive noise \mathbf{n}

$$\mathbf{v}[i, j] = \mathbf{h} + \mathbf{n}[i, j] \quad (9)$$

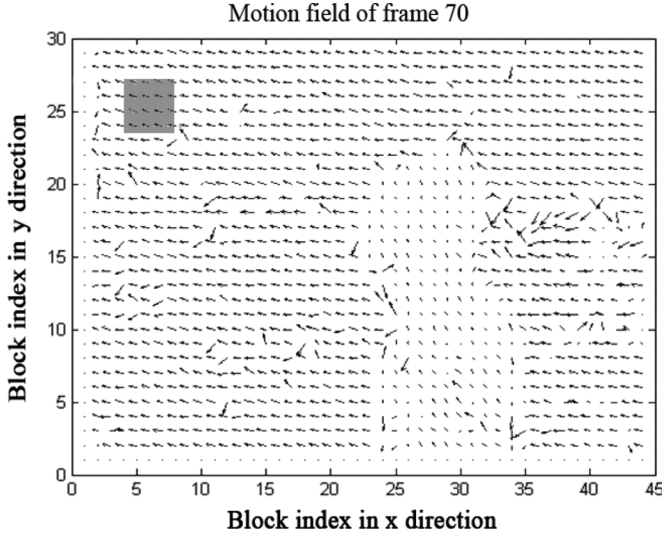


Fig. 15. Background region (indicated by the gray area) extracted from a motion field.

where (i, j) is the index of the MB and $\mathbf{n}[i, j]$ is assumed to be independent and identically distributed (i.i.d.) Gaussian signal with zero-mean. With this model, the best BR is determined by

$$\text{BR} = \arg \min_{(i,j)} \left\{ \sum_{k=-w}^{w-1} \sum_{l=-w}^{w-1} |\mathbf{v}[i+k, j+l] - \mathbf{h}| \right\} \quad (10)$$

where w is an adjustable parameter that controls the number of MBs for the BR. Note that the BR of each frame is dynamically estimated, and the size of the BR does not have to be fixed either. An example of BR is shown in Fig. 15.

The BR determined from frame $n-1$ is used as the prediction of the BR of frame n . Therefore, after the BR is identified, the digital stabilizer performs BME on all MBs within the BR for the next input frame. In our simulation, we set the size of the BR to as small as 64×64 (i.e., $w = 2$), and the resulting GM is still very accurate. At this BR size, the computational cost of BME for the digital stabilizer is reduced by a factor of $(720 \times 480)/(64 \times 64) = 84,375$.

The system flow shown in Fig. 14(a) is designed for the case where the structure of the group of pictures (GOP) only includes I- and P-frames. When the GOP structure contains B-frames (e.g., IBBP), the frames are not encoded in the same order as they are received. Since the B-frames are encoded after the P-frame, we cannot use the motion data of the P-frame to stabilize the B-frames. Furthermore, it becomes complicated to use the motion data of a B-frame to determine its BR. In light of these reasons, BR identification is not performed for B-frames; thus the GM detection for determining \mathbf{h} (and hence the BR) is skipped. Also, instead of using the BR from the previous frame, the digital stabilizer performs BME on the BR identified in the closest P-frame. The assignments of BR for the two GOP structures considered here are shown in Fig. 16.

Compared to the IPPP structure, the duration between two neighboring P-frames becomes longer when the GOP structure is IBBP, and hence the accuracy of BR may suffer. This problem can be alleviated by adjusting the control parameter w in (9) to increase the BR size.

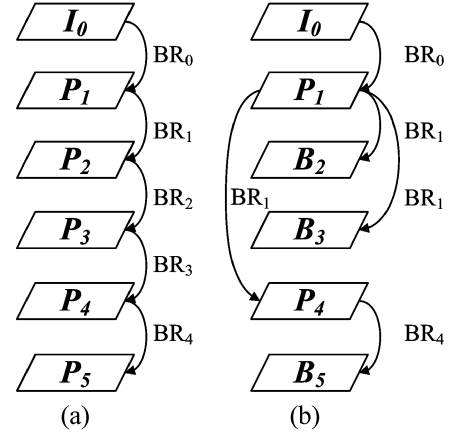


Fig. 16. Assignment of BR under different GOP structures. (a) IPPP. (b) IBBP.

B. Scheme 2

In Scheme 2, the digital stabilizer computes the integer-pel MVs of the MBs of each video frame. The resulting MVs are input to the video encoder as the initial motion predictions. Then the video encoder performs fast search and subpixel ME to refine the MVs.

The major challenge of this scheme is the misalignment of MBs before and after image stabilization. The stabilized video frame input to the video encoder is a displaced version of the original video frame. Therefore, the block decomposition of the video frame to be coded is no longer in alignment with that of the original frame. This situation is illustrated in Fig. 17(a), where MB A of a stabilized frame contains part of MBs 1, 2, 3, and 4 of the original frame. As a result, each of the four MVs generated by the digital stabilizer for MBs 1, 2, 3, and 4 only corresponds to part of MB A. Consequently, the video encoder cannot simply take any of these MVs as the motion prediction for MB A.

As a starting point, the MV of MB A can be estimated from the four MVs by interpolation. This is illustrated in Fig. 17(b), where \mathbf{v}_i denotes the MV of MB i , $i = 1, 2, 3, 4$. Then the MV \mathbf{v}_A of MB A is obtained by

$$\mathbf{v}_A = \left(\frac{1}{256} \right) \sum_{i=1}^4 w_i \mathbf{v}_i - \mathbf{d}_n + \mathbf{d}_{n-1} \quad (11)$$

where w_i is a weighting factor that is equal to the number of pixels inside the intersection of MBs i and A, and \mathbf{d}_n and \mathbf{d}_{n-1} , respectively, are the displacement vectors of the current and previous stabilized frames, as defined by (7). Here, the default size of a MB is 16×16 .

In general, the interpolated MVs obtained by (11) are sufficiently close to the ones obtained by full-search ME (FSME) if it were used to estimate the motion. However, such an interpolation scheme gives poor results for regions across the boundary of moving objects. Any further local refinement of the MVs of such regions cannot help improve the accuracy. The coding efficiency drops.

To solve this problem, we do the motion interpolation selectively. More specifically, we measure the variance of the four MVs \mathbf{v}_i used in (10). If the variance is small enough, the video

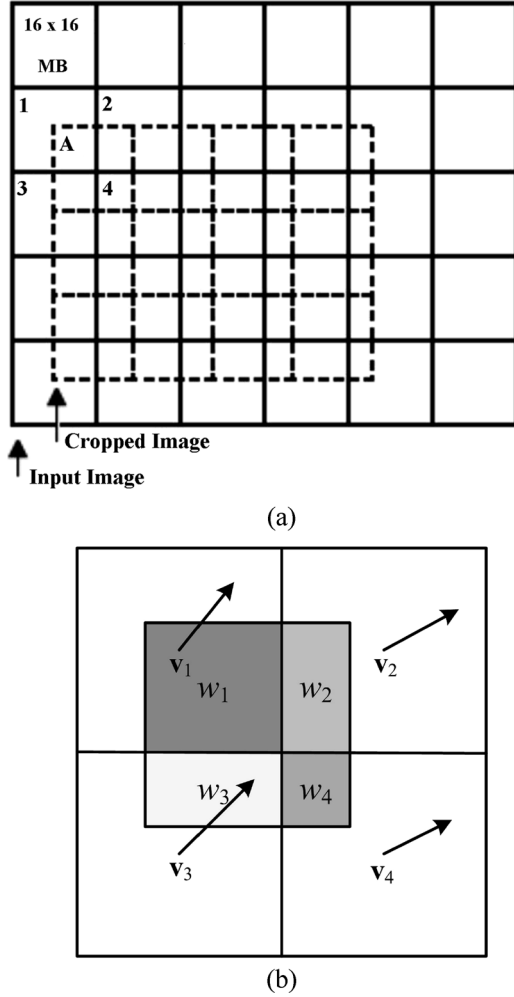


Fig. 17. (a) Misalignment of MBs between the images before and after stabilization. (b) Motion interpolation in Scheme 2.

encoder interpolates the MVs according to (10) and then applies a small diamond search (SDS) [27] around the interpolated MV to improve the result. Otherwise, the video encoder applies FSME to obtain the MV for MB A. In both cases, a subpixel ME is applied to further refine the resulting integer-pel MVs. With this approach, the coding efficiency is greatly improved. We find from our experiments that only 13% of the MBs require FSME in the video encoder. For the remaining MBs, the video encoder does not have to perform FSME. MVs with good accuracy can be obtained by simply applying motion interpolation and SDS for these MBs.

The system flow of the scheme is depicted in Fig. 18(a). For simplicity only the function blocks that are involved in the integration are shown. Here, like in Fig. 14, the white blocks represent the functions performed by the digital stabilizer, the dark gray blocks are the functions performed by the video encoder, and the gray block is the new function introduced as a result of the integration. The data flow of the scheme is shown in Fig. 18(b).

C. Scheme 3

In this scheme, the digital stabilizer is integrated with the video decoder, and the compressed video is stabilized while

being decoded. The system flow is shown in Fig. 19. Because the motion information of the video sequence already exists in the video bit stream, the digital stabilizer can simply apply the decoded MVs to perform GM detection and smoothing by following the procedure described in Section II-B.

However, the motion data available in the bit stream may not correspond to the motion between neighboring frames required for digital stabilization. For example, when the GOP structure is IBBP, the motion of the P-frame is measured with respect to the I-frame. But what is needed for stabilizing this P-frame is the motion relative to the preceding B-frame in the display order. The main task of Scheme 3, therefore, is to derive such motion from the available motion data.

To simplify the computation, each available MV is converted to the required MV by either linear interpolation or extrapolation, depending on whether the available MV represents a forward MV or a backward MV. The resulting MVs are referred to as regularized MV in this paper. All MVs are converted to forward MVs between two temporally neighboring frames. Continue the previous example. The frame distance between I- and P-frames is 3. Thus, each MV of the P-frame is linearly interpolated by scaling down the MV decoded from the bit stream by a factor of 3. Likewise, the forward MV of a MB in a B-frame is converted from a backward MV of the B-frame by changing the sign of the backward MV and normalizing its value by the frame distance between the B-frame and the reference frame based on which the backward MV is estimated. The formula for converting the available MVs to the regularized MVs are listed in Table V.

V. PERFORMANCE EVALUATION

The performance of the three integration schemes described in the previous section is evaluated in three aspects: computational cost, coding efficiency, and accuracy of GM estimation.

A. Computational Cost

Experiments are performed to compare the computational cost between the integrated systems and the original nonintegrated system. The computational cost is measured in terms of the average number of integer-pel FSMEs per frame. The frame size is 720×480 before stabilization and 688×448 after stabilization, and the search range is 32. We do not use any speed-up technique (such as early termination) for FSME; therefore, the computational cost we obtain is independent of the video content, and the reduction is purely due to the integration. For comparison purpose, the computational cost of each scheme is normalized with respect to the computational cost of the pure video encoder with input image size 688×488 .

The results are shown in Table VI. As we can see, the computational cost of a system with separate digital stabilizer and video encoder is twice more than that of a pure video.

For Scheme 1, the computational cost depends on the size of the BR [w in (9)]. Among the three cases considered for this scheme, $w = 2$ gives reasonably good MVs while $w = 8$ generates superb result. Comparing with a pure video encoder, the additional computational cost introduced by Scheme 1 ranges from 1% to 21%.

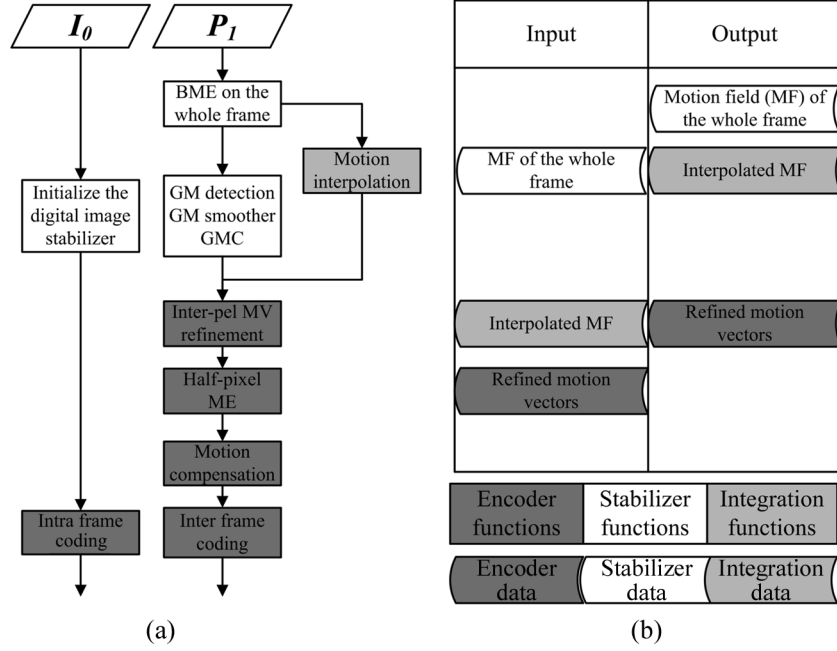


Fig. 18. System flow of Scheme 2. (a) Operation of the first two frames I_0 and P_1 . (b) Input and output data at each stage. Only functions that are related to the integration are shown.

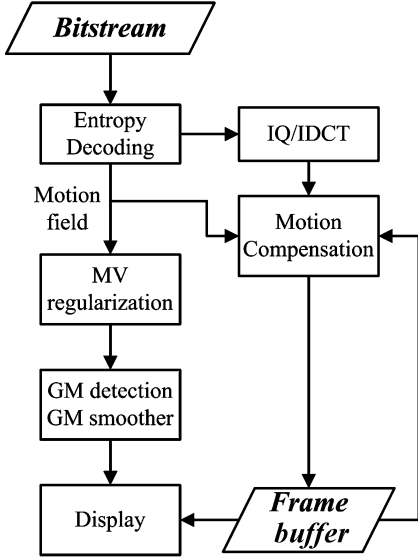


Fig. 19. System flow of Scheme 3.

TABLE V
REGULARIZATION OF MVs IN SCHEME 3

MB mode /Frame type	Reference frame(s)	Regularized MV
Intra/ I_n , P_n , and B_n	None	$\text{average}(g[n-1], g[n+1])$
Inter/ P_n	$P(I)_{n-2}$	$\mathbf{v}/3$
Forward/ B_n	$P(I)_{n-1}$ $P(I)_{n-2}$	\mathbf{v} $\mathbf{v}/2$
Backward/ B_n	$P(I)_{n+1}$ $P(I)_{n+2}$	$-\mathbf{v}$ $-\mathbf{v}/2$
Bi-directional, Direct / B_n	$\{P(I)_{n-1}, P(I)_{n+2}\}$ $\{P(I)_{n-2}, P(I)_{n+1}\}$	$\text{average}(\mathbf{v}_F, -\mathbf{v}_B/2)$ $\text{average}(\mathbf{v}_F/2, -\mathbf{v}_B)$

GOP structure is IBBP.

\mathbf{v}_F denotes the forward MV, and \mathbf{v}_B the backward MV.

In Table VI, the computational cost of Scheme 2 is shown for different cases, depending on the methods used for the refinement of the interpolated MVs. If only subpel refinement is performed around the interpolated MV without small diamond search, the computational cost increases about 12%. This increment due to larger input frame for stabilizer. If the small diamond search is applied, the computational cost is increased by 14%. Here, the cost of SDS is converted to the cost of FSME by considering the number of SADs calculated. We evaluate this cost on various sequences and find the variance of the increased cost is small. Therefore, we believe it can be regarded content-independent. In the final case, we apply SDS in regions where the variance of MVs is small and FSME in regions where the variance is large. The computational cost increases by 16%, which is still much smaller than that of the nonintegration system.

Scheme 3 integrates the digital stabilizer with the video decoder and the BME operation is totally skipped in the digital stabilizer. Compared with the operations in the video decoder, the cost of GM detection and GM smoothing in the digital stabilizer is less than 1%.

B. Coding Efficiency

In Scheme 1, the additional functions, GM detection and BR identification, introduced to the integrated system are performed after video encoding. Recall that in Section III we show that video stabilization has little effect on coding efficiency. It can be expected that a slight change in the way the GM is computed, as in the integration scheme, would have little effect on the coding efficiency.

In Scheme 2, the ME process of the video encoder is modified as a result of the integration. We implement Scheme 2 on the C++ MPEG-4 reference software to investigate the impact of

TABLE VI
COMPARISON OF COMPUTATIONAL COST

Integration Systems	#FSME in the image stabilizer	#FSME in the video encoder	Total #FSME for one frame	Normalized computational cost (%)
Standalone image stabilizer and video encoder	1350	1204	2554	212.13
Scheme 1, $w=2$	16	1204	1220	101.33
Scheme 1, $w=4$	64	1204	1268	105.32
Scheme 1, $w=8$	256	1204	1460	121.26
Scheme 2, without SDS	1350	0	1350	112.13
Scheme 2, with SDS	1350	23.5	1373.5	114.08
Scheme 2, with SDS and variance checking	1350	46.57	1396.57	115.99
Pure video encoder with input size 720×480	0	1350	1350	112.13
Pure video encoder with input size 688×448	0	1204	1204	100.00

#FSME: Number of full-search MEs performed in one frame.

The cost of SDS is converted to #FSME according to the average checked candidates. Input frame size of the image stabilizer = 720×480 . Input frame size of the video encoder = 688×448 .

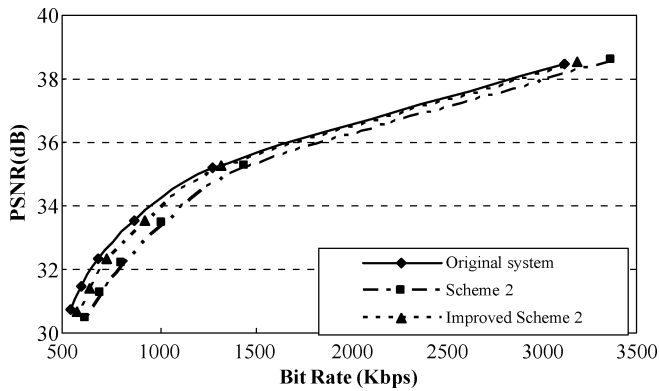


Fig. 20. R-D curves of the sequence *Basketball* encoded by MPEG-4 SP encoder.

integration on the coding efficiency. The rate-distortion (R-D) curves of the sequence *Basketball* are shown in Fig. 20.

We compare two cases. The curve that is labeled “Scheme 2” in Fig. 20 corresponds to the case in which the video encoder performs half-pixel MV refinement around the interpolated MVs. The coding performance of this scheme decreases about 0.1 dB at high bit rates and 0.8 dB at low bit rates. As explained earlier, this is due to the inaccuracy of the interpolated MVs for regions around the boundaries of objects.

The curve that is labeled “Improved Scheme 2” in Fig. 20 represents the second case in which the motion interpolation is performed selectively according to the variance of the four reference MVs. Because the interpolated MV is refined using the small diamond search and new MV are computed using FSME for MBs with large motion variance, the coding performance is very close to that of the original video encoder. The improvement in coding efficiency over the first case is consistent across all bit rates.

Scheme 3 is an integration of the digital stabilizer and the video decoder. Therefore, it does not affect the coding efficiency of the video encoder. However, the required bit rate for the video will inevitably increase because the encoded frame must be larger than the display window.

C. Accuracy of Global Motion Detection

In Scheme 1, the digital stabilizer only performs BME in the BR. With $w = 2$, the GMVs determined by the original dig-

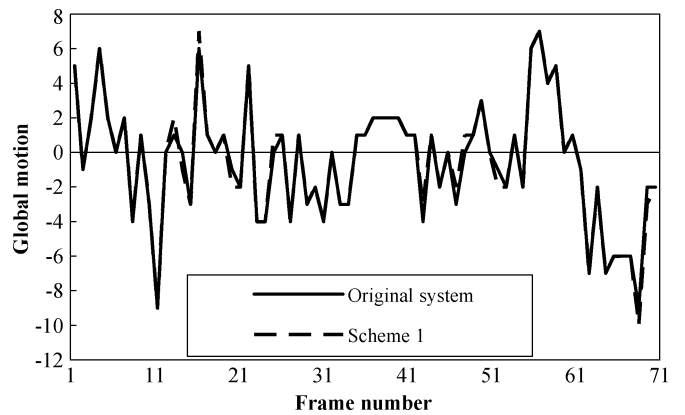


Fig. 21. Comparison of GMVs estimated by Scheme 1 and the original image stabilizer. Only the y -component is depicted here.

ital stabilizer and this integrated scheme are shown in Fig. 21. The RMSE of the difference between these two GMV curves is 0.034 only. With $w = 8$, this RMSE value reduces to 0.014. In practice, we have found that $w = 2$ is generally good enough. Choosing an extremely small value for w is not recommended since it is more likely to run into detection errors caused by fast moving objects or textureless areas.

In Scheme 2, we do not modify the image stabilizer, so the accuracy of GMVs remains the same as that of the original digital stabilizer.

In Scheme 3, the GM is estimated from the motion information in the bit stream. Our simulation shows that the results are very accurate. For example, for the *Basketball* sequence the RMSE between GMVs generated by Scheme 3 and the original stabilizer is 0.033.

VI. DISCUSSION

The design of the digital stabilizer and the integration schemes described in the previous sections is the result of a tradeoff between complexity, codec compatibility, and performance. These issues are further discussed in this section.

A. Higher Order Global Motion Models

To be consistent with the video encoder, the translational motion model is considered for image stabilization in this work. In addition to this simple model, higher order motion models

that take, for example, zooming and rotational parameters into account have been adopted for GM estimation [30]–[32]. The method described in [30] estimates a higher order motion model by calculating the directional derivatives of the motion field, while the method described in [31] and [32] uses a maximum likelihood method to estimate the GM from the local motion field.

In principle, incorporating such higher order motion models into the integration framework is not a problem. By generalizing the BR identification and motion interpolation, the digital stabilizer and the video encoder can be integrated nicely.

However, higher order GM models lead to a more elaborated GM compensation process that involves computationally intensive operations such as image warping. Such operations are prohibitive for portable devices that are sensitive to power consumption. From the hardware implementation standpoint, such operations cause the system flows of the stabilization and non-stabilization modes of a digital video camera to become quite different and difficult to integrate with the block-level pipeline architectures of most video encoders.

In summary, the simple translational motion model works effectively well and is good enough for many applications. If a higher order motion model is to be used for image stabilization, the performance gain has to be high enough to justify the additional cost incurred due to the use of higher order motion models.

B. Search Range, Resolution, and Fast Motion Estimation

We find that incorrect GMVs would result if the true GM is larger than the search range. For video encoder, incorrect MVs only affect the coding performance; the visual quality can be maintained at the cost of a higher bit rate. For digital stabilizer, however, incorrect GMVs result in poor stabilization performance and even introduce abrupt jiggles to the image sequence. Therefore, large search window for the BR is preferable for Scheme 1. Since the BR is relatively small, the additional computational cost due to the use of large search windows is negligible.

Our digital stabilizer computes integer-pel MVs. Although half-pel and quarter-pel GMVs can be computed, it involves interpolating the whole image. Therefore, the same computational issue as discussed for higher order motion model arises, leaving the payoff in question. In addition, the coding performance will be inevitably affected because the input to the video encoder is an interpolated (filtered) sequence.

Fast ME for video encoding has been extensively studied. The integration framework presented here does not prevent the use of any such fast ME algorithm for digital stabilization, but it has to be done carefully since the accuracy of the GM can be affected.

C. Full Frame Recovery

In practice, one should choose an image sensor that is large enough to allow the digital stabilizer to function well. Due to GM compensation, the actual size of the stabilized video frame becomes smaller than the input video frame. Alternatively, if we choose to maintain the stabilized video at the original size, it will contain undefined pixels. In the former case, the image margin

remaining after cropping dictates the extent that the camera motion can be compensated. A bigger image margin means a larger camera shaking can be handled, but the displayable image area becomes smaller. To restore the video to its original size, one may apply an up-sampling filter to the stabilized video. In the latter case, video completion techniques such as mosaicing [28] or inpainting [29] can be applied to fill in the undefined image area.

All these full frame recovery techniques can be incorporated into our integration schemes to form an integral part of the whole system.

VII. CONCLUSION

In this paper, we have presented a digital stabilization algorithm and three integration schemes for integrating the digital stabilizer with the video codec of digital video cameras. Designed by taking hardware architecture into consideration, the digital stabilizer works as effectively as gyro-based sensors and integrates nicely with the video codec. The GM determined by these integration schemes is as accurate as that by the standalone digital image stabilizer, and the coding performance is as good as the standalone video encoder. As a result of the integration, Schemes 1 and 2 reduce the computational cost by a factor of 2, and Scheme 3 costs almost nothing as compared to the video decoder.

ACKNOWLEDGMENT

The authors would like to thank Mr. Kuo for developing the camera testbed described in Section II-C.

REFERENCES

- [1] M. Oshima, T. Hayashi, S. Fujioka, T. Inaji, H. Mitani, J. Kajino, K. Ikeda, and K. Komoda, "VHS camcorder with electronic image stabilizer," *IEEE Trans. Consum. Electron.*, vol. 35, no. 4, pp. 749–758, Nov. 1989.
- [2] K. Sato, S. Ishizuka, A. Nikami, and M. Sato, "Control techniques for optical image stabilizing system," *IEEE Trans. Consum. Electron.*, vol. 39, no. 3, pp. 461–466, Aug. 1993.
- [3] K. Uomori, A. Morimura, H. Ishii, T. Sakaguchi, and Y. Kitamura, "Automatic image stabilizing system by full-digital signal processing," *IEEE Trans. Consum. Electron.*, vol. 36, no. 3, pp. 510–519, Aug. 1990.
- [4] J.-K. Paik, Y.-C. Park, and S.-W. Park, "An edge detection approach to digital image stabilization based on tri-state adaptive linear neurons," *IEEE Trans. Consum. Electron.*, vol. 37, no. 3, pp. 521–530, Aug. 1991.
- [5] J.-K. Paik, Y.-C. Park, and D.-W. Kim, "An adaptive motion decision system for digital image stabilizer based on edge pattern matching," *IEEE Trans. Consum. Electron.*, vol. 38, no. 3, pp. 607–616, Aug. 1992.
- [6] Y. Egusa, H. Akahori, A. Morimura, and N. Wakami, "An application of fuzzy set theory for an electronic video camera image stabilizer," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 351–356, Aug. 1995.
- [7] S.-J. Ko, S.-H. Lee, and K.-H. Lee, "Digital image stabilizing algorithms based on bit plane matching," *IEEE Trans. Consum. Electron.*, vol. 44, no. 3, pp. 544–549, Aug. 1998.
- [8] S.-J. Ko, S.-H. Lee, and S.-W. Jeon, "Fast digital image stabilizer based on gray-coded bit plane matching," in *Proc. Int. Conf. Consum. Electron.*, Los Angeles, CA, Jun. 1999, pp. 90–91.
- [9] G.-R. Chen, Y.-M. Yeh, S.-J. Wang, and H.-C. Chiang, "A novel structure for digital image stabilizer," in *Proc. Asia-Pacific Conf. Circuits Syst.*, Dec. 2000, pp. 101–104.
- [10] H. R. Pourreza, M. Rahmati, and F. Behazin, "Weighted multiple bit plane matching, a simple and efficient matching criterion for electronic digital image stabilizer application," in *Proc. Int. Conf. Signal Process.*, Aug. 2002, vol. 2, pp. 957–960.
- [11] S. Ertürk, "Image sequence stabilisation by lowpass filtering of inter-frame motion," in *Proc. SPIE, Vis. Commun. Image Process.*, 2001, vol. 4310, pp. 434–442.

- [12] S. Ertürk and T. J. Dennis, "Image sequence stabilisation based on DFT filtering," *IEE Proc. Image Vis. Signal Process.*, vol. 127, pp. 95–102, 2000.
- [13] S. Ertürk and M. K. Gullu, "Membership function adaptive fuzzy filter for image sequence stabilization," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 1–7, Feb. 2004.
- [14] S. Ertrük, "Real-time digital image stabilization using Kalman filters," *Real-Time Imaging*, vol. 8, pp. 317–328, 2002.
- [15] S.-C. Hsu, S.-F. Liang, and C.-T. Lin, "A robust digital image stabilization technique based on inverse triangle method and background detection," *IEEE Trans. Consum. Electron.*, vol. 51, no. 2, pp. 335–345, May 2005.
- [16] H. R. Pourreza, M. Rahmati, and F. Behazin, "An electronic digital image stabilizer based on stationary wavelet transform (SWT)," in *Proc. Int. Conf. Image Process.*, Barcelona, Spain, Sep. 2003, vol. 2, pp. 383–386.
- [17] C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Seattle, WA, May 1998, vol. 5, pp. 2789–2792.
- [18] S. B. Balakirsky and R. Chellappa, "Performance characterization of image stabilization algorithms," in *Proc. Int. Conf. Image Process.*, Sep. 1996, vol. 1, pp. 413–416.
- [19] A. Engelsberg and G. Schmidt, "A comparative review of digital image stabilising algorithms for mobile video communications," *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 591–597, Aug. 1999.
- [20] S. Ertürk, "Digital image stabilization with subimage phase correlation based global motion estimation," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1320–1325, Nov. 2003.
- [21] J.-Y. Chang, W.-F. Hu, M.-H. Cheng, and B.-S. Chang, "Digital image translational and rotational motion stabilization using optical flow technique," *IEEE Trans. Consum. Electron.*, vol. 48, no. 1, pp. 108–115, Feb. 2002.
- [22] A. Censi, A. Fusiello, and V. Roberto, "Image stabilization by feature tracking," in *Proc. Int. Conf. Image Anal. Process.*, Venice, Italy, Sep. 1999, pp. 665–667.
- [23] A. J. Crawford, H. Denman, F. Kelly, F. Pitie, and A. C. Kokaram, "Gradient based dominant motion estimation with integral projections for real time video stabilization," in *Proc. Int. Conf. Image Process.*, Singapore, Oct. 2004, vol. 5, pp. 3371–3374.
- [24] A. T. Erdem and C. Erdem, "Effect of image stabilization on the performance of the MPEG-2 video coding algorithm," in *Proc. SPIE, Vis. Commun. Image Process.*, 1998, vol. 3309, pp. 272–277.
- [25] *Generic Coding of Audio-Visual Objects—Part 2 Visual*, ISO/IEC 14496-2, 2001.
- [26] *Advanced Video Coding, Draft ITU-T Recommendation H.264 and Final Draft International Standard of Joint Video Specification*, 14496-10, May 2003.
- [27] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, no. 2, pp. 287–290, Feb. 2000.
- [28] A. Litvin, J. Konrad, and W. C. Karl, "Probabilistic video stabilization using Kalman filtering and mosaicing," in *Proc. IS&T/SPIE Symp. Electron. Imaging, Image Video Commun.*, 2003, pp. 663–674.
- [29] Y. Matsushita, E. Ofek, X. Tang, and H. Y. Shum, "Full-frame video stabilization," in *Proc. Int. Conf. Comp. Vis. Pattern Recognit.*, San Diego, CA, Jun. 2005, vol. 1, pp. 50–57.
- [30] R. Coudray and B. Besserer, "Global motion estimation for MPEG-encoded streams," in *Proc. Int. Conf. Image Process.*, Singapore, Oct. 2004, pp. 3411–3414.
- [31] Y. Su, M.-T. Sun, and V. Hsu, "Global motion estimation from coarsely sampled motion vector field and the applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 232–242, Feb. 2005.
- [32] Y. Su, M.-T. Sun, and V. Hsu, "Global motion estimation from coarsely sampled motion vector field and the applications," in *Proc. Int. Symp. Circuits Syst.*, May 2003, vol. 2, pp. 628–631.
- [33] Y.-C. Peng, H.-A. Chang, H. H. Chen, and C.-J. Kao, "Digital image stabilization and its integration with video encoder," in *Proc. Consum. Commun. Netw. Conf.*, Las Vegas, NV, Jan. 2005, pp. 617–622.
- [34] C.-K. Liang, Y.-C. Peng, H.-A. Chang, C.-C. Su, and H. H. Chen, "The effect of digital image stabilization on coding performance," in *Proc. Int. Symp. Intell. Multimedia, Video Speech Process.*, Hong Kong, Oct. 2004, pp. 402–405.

- [35] Y.-C. Peng, C.-K. Liang, H.-A. Chang, H. H. Chen, and C.-J. Kao, "Integration of image stabilizer with video codec for digital video cameras," in *Proc. Int. Symp. Circuits Syst.*, Kobe, Japan, May 2005, pp. 4781–4784.
- [36] Y.-C. Peng, M.-T. Lu, and H. H. Chen, "DSP implementation of digital image stabilizer," in *Proc. Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, Jul. 2005, pp. 808–811.



Homer H. Chen (S'83–M'86–SM'01–F'03) received the Ph.D. degree in electrical and computer engineering from University of Illinois at Urbana-Champaign.

Since August 2003, he has been with the College of Electrical Engineering and Computer Science, National Taiwan University, Taiwan, R.O.C., where he is Irving T. Ho Chair Professor. Prior to that, he had held various research and development management and engineering positions in leading US companies including AT&T Bell Labs, Rockwell Science Center, iVast, and Digital Island over a period of 17 years. He was a US delegate of the ISO and ITU standards committees and contributed to the development of many new interactive multimedia technologies that are now part of the MPEG-4 and JPEG-2000 standards. His research interests lie in the broad area of multimedia processing and communications.

Dr. Chen is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He served as Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING from 1992 to 1994, Guest Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 1999, and Editorial Board Member for *Pattern Recognition* from 1989 to 1999.



Chia-Kai Liang (S'05) was born in Kaohsiung, Taiwan, R.O.C. He received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, R.O.C., in 2004, where he is currently working toward the Ph.D. degree in the Graduate Institute of Communication Engineering.

His research interests include image/video processing, video coding, and digital rights management.



Yu-Chun Peng was born in Miaoli, Taiwan, R.O.C., in 1981. He received the B.S. degree in electrical engineering and the M.S. degree in 2005 from the Graduate Institute of Communication Engineering, National Taiwan University, Taiwan, R.O.C., in 2003 and 2005, respectively.

Currently, he is working in Novatek Inc., Taiwan, R.O.C. His research interests are in the area of image and video processing for digital still camera and video camera.



Hung-An Chang was born in Taipei, Taiwan, R.O.C. He received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, R.O.C., in 2004. He is currently working toward the Ph.D. degree in the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

His research interests are in the area of digital signal processing, communication, image/speech processing, and video coding.