Digital Image Stabilization and Its Integration with Video Encoder

Yu-Chun Peng, Hung-An Chang, Homer H. Chen
Graduate Institute of Communication Engineering
National Taiwan University
Taipei, Taiwan
{b88901089, b89901049}@ntu.edu.tw, homer@cc.ee.ntu.edu.tw

Chang-Jung Kao
Quanta Research Institute
Quanta Computer Inc.
Taipei, Taiwan
chris.kao@quantatw.com

Abstract—**Digital image stabilizer and video encoder are two important components of a digital video camera. The digital image stabilizer compensates the image movement caused by hand jiggles and thereby improves the perceptual quality of the captured image sequence, while the video encoder compresses the huge amount of video data down to a reasonable size. Both components require the motion information of the image sequence to perform their respective tasks. Since motion estimation is a computation intensive operation, we present an integration scheme for integrating the digital image stabilizer with the video encoder. The image stabilization algorithms and the technical issues involved in the integration are discussed. Simulation results are shown to illustrate the effectiveness of the proposed digital image stabilization system.**

I. Introduction

The movement of a video camera caused by hand jiggle introduces jerky image motion that is often annoying to human eyes. The problem can be solved by using an image stabilizer to compensate for the hand motion. As opposed to optical image stabilization which uses gyro-sensors to detect hand motion and shifts a corrective lens inside the lens system (or alternatively shifts the image sensor while keeping the lens fixed) , digital image stabilization detects the induced image motion based on the video data and shifts the image display window accordingly to compensate for the hand motion. For many digital video camera designs, digital image stabilization is a cost effective solution. The digital image stabilization problem has been studied in the past. Uomori et al. [1] proposed a full-digital signal processing stabilization system that estimates the global motion by correlating block-based local motion vectors. Paik et al. [2] [3] proposed the estimation of global motion from the isolativity and stability of local motion vectors determined by edge-pattern matching. In [4], a lowpass filter is used to smooth interframe motion. A comparative review of the stabilization for mobile video communications is presented in [5].

For videos captured by digital video cameras, hand jiggles or camera panning introduces a global motion between successive frames. However, the rate of hand jiggling is much higher than that of camera panning. Thus the induced global motion can be lowpass filtered temporally to reduce the effect of hand jiggling on the visual appearance of the sequence and to smooth the camera panning. The image stabilization is accomplished by moving the display window within the original image boundary to compensate the part of image movement introduced by hand jiggling.

In a digital video camera, both the video encoder and the image stabilizer need to compute the image motion, but motion estimation is a computation intensive operation. Therefore, it makes sense from the system design point of view to integrate the motion estimation modules of these two components. The challenge is that the accuracy requirements of the two components are different. While the performance of the video encoder is highly sensitive to the accuracy of estimated motion vectors, the image stabilizer cares more about accurate classification of motion vectors into camera/object motions. The granularity requirements of motion field are different too. The video encoder demands one motion vector per block, whereas the image stabilizer simply needs to compute one induced global motion vector per frame. In our integrated scheme, the image stabilizer reduces the computational cost by confining the global motion estimation to within the background region determined by the video encoder.

This paper is organized as follows. We describe the architecture of the digital image stabilizer in Section 2 and discuss the algorithms developed for stabilizing camera zooming and panning in section 3. In section 4, in order to integrate the digital image stabilizer with video encoder, we setup the experiments for testing the effect of digital image stabilization on the performance of two video encoders, MPEG-4 and H.264. Then the integration scheme is described in section 5, followed by a summary.

II. System Architecture

Fig. 1 shows the architecture of the digital image stabilizer we have developed. First, the motion field between two successive frames is computed by block-based motion estimation, as in most video coding techniques. The resulting motion estimates are input to the global motion estimation module to determine the global motion by clustering. Then a motion smoother (a lowpass filter) is applied to the global motion frame by frame to remove the unstable camera movements. Finally, the current frame is motion compensated by shifting the display window according to the difference between the smoothed global motion and the original one. The stabilized image sequence is then input to the video encoder.

This work was supported in part by the National Science Council of Taiwan under contract NSC93-2752-E-002-006-PAE.

0-7803-8784-8/04/$20.00 © 2004 IEEE

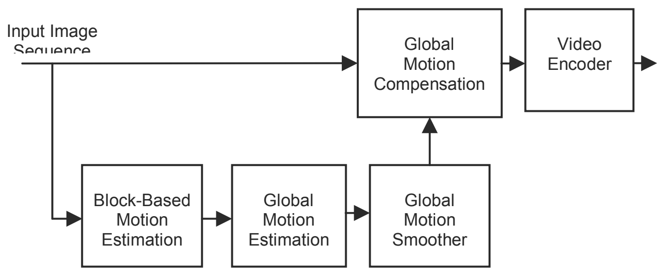


Figure 1. The block diagram of the image stabilizer.

A. *Global Motion Estimation*

Typically, the motion of background blocks is caused by the camera motion. As long as the scene is not dominated by one single moving object, the cluster corresponding to background blocks has the maximum votes in the clustering process. The average of this cluster of motion vectors is chosen as the global motion.

An example is given in Fig. 2, where the motion field between frame 69 and 70 of Sequence 2 is shown. The camera motion can be well approximated by using the motion vectors of the background blocks. Fig. 3 shows the clustering of a local motion field. The motion vectors are clustered into several groups. In this case, the group located at (-22, -4) receives the maximum votes and becomes the winner of the global motion estimation.

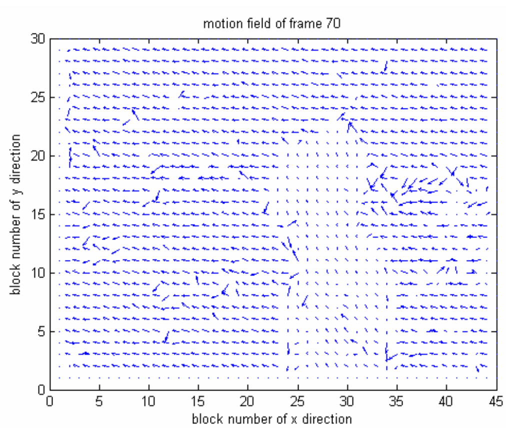


Figure 2. Motion field of frame 70.

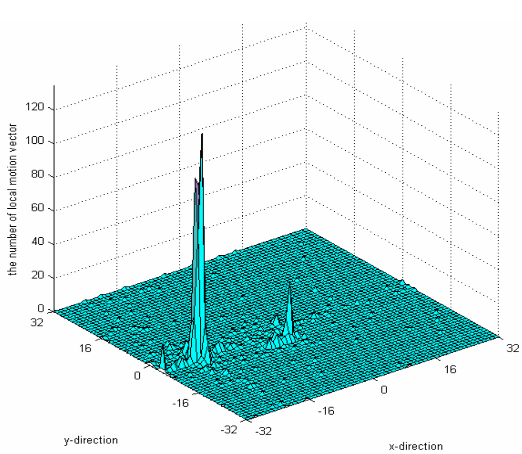


Figure 3. Clusters of a motion field.

B. *Global Motion Smoother*

We apply a lowpass filter similar to that in [4] to smooth the global motion. Under the assumption that the frequency of the unstable movement is much higher than that of the intentional camera motion, the motion smoother removes the high frequency components of the global motion. The digital filter is described by,

$$G_s[n] = \sum_{i=0}^{m} a_i * G[n-i] \tag{1}$$

$$AG_s[n] = AG_s[n-1] + G_s[n] \tag{2}$$

where G[n] represents the global motion at Frame n, $G_s[n]$ the resulting smoothed motion, AG[n] the accumulated global motion and a_i's the normalized coefficients of the filter.

To allow for real-time processing, we use a causal lowpass filter. Eq. (1) can be considered as a weighted moving average filter with window size m. G_s gets smoother as m increases. Fig. 4 shows the curves of G and G_s in the y-direction for m=7. The corresponding accumulative global motions with respect to the first frame are shown in Fig. 5. As we can see, the stabilization effect is achieved.

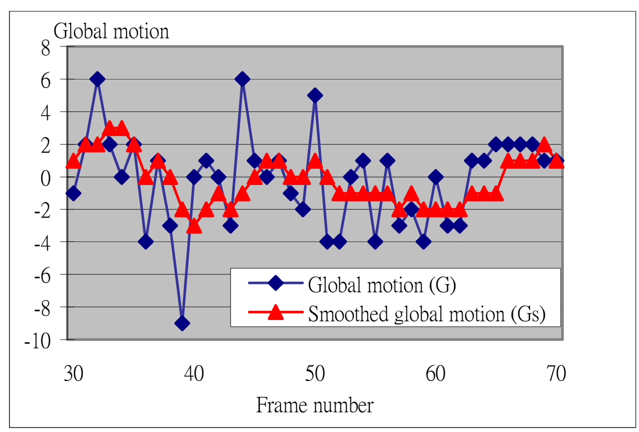


Figure 4. The global motion before and after smoothing.

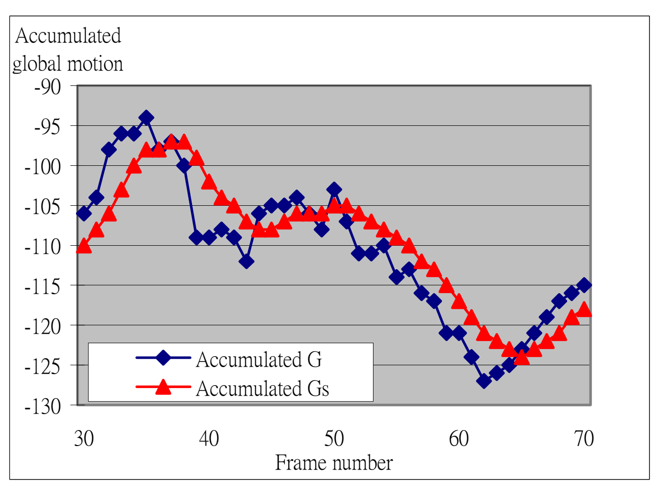


Figure 5. Accumulated global motion before and after smoothing.

III. IMAGE STABILIZATION FOR CAMERA ZOOMING

Optical zooming is an important function of digital video cameras. An optical zooming introduces a global motion on the image plane. Fig. 6 shows the local motion field as the result of a pure optical zooming with no camera movement. Note that the motion field appears radial and that the center of the motion field is near zero.

A. Camera zooming

In the case where both camera motion and optical zooming occur, the resulting local motion field would look like the one shown in Fig. 7. Unlike the pure zooming case, the local motion vectors at the center of the motion field are not zero. These nonzero motion vectors are the result of camera movement. In our algorithms, they serve as the basis for computing the camera movement. However, dividing the central region into small blocks and applying block-based motion estimation to each block may lead to inaccurate global motion estimation. To solve the problem, we select a large block (240x320) at the central region of the image and use it to estimate the global motion by block matching:

$$SAD(u,v) = \sum_{(x,y)\in block} \left|F_n(x,y) - F_{n-1}(x+u, y+v)\right| \quad \textbf{(3)}$$

where $F_n(x,y)$ is a block in the n^{th} frame. Within the search window, the displacement (u,v) that gives rise to the minimum SAD (sum of absolute difference) is solution for the global motion. Then, as described in section 2, the stabilized image sequence is obtained by smoothing the global motion of each frame. Normally, a larger search window leads to a better estimate of the global motion but at the expense of additional computations.

B. Detection of rapid camera panning

The image frames are repositioned to achieve stabilization by shifting the display window according to the difference between the original global motion (G) and the smoothed global motion (G_s). As rapid panning occurs, we want the display window to follow the camera motion as closely as possible. While being able to removes the unstable movement, the lowpass filter described in Section 2.B introduces a delay to the stabilized image sequence. The apparent delay between the stabilized image sequence and the original sequence increases as the camera pans rapidly. To solve the problem, a fuzzy control approach is proposed in [9]. Here, we use a weighted average approach, in which the smoothed global motion is refined as follows:

$$G_s^{'}[n] = b * G_s[n] + (1-b) * G[n] \quad \textbf{(4)}$$

This helps the display window to catch up with camera panning.

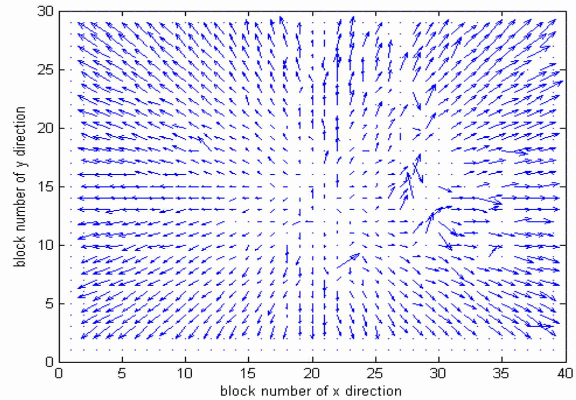


Figure 6. Motion field of zooming without camera movement.

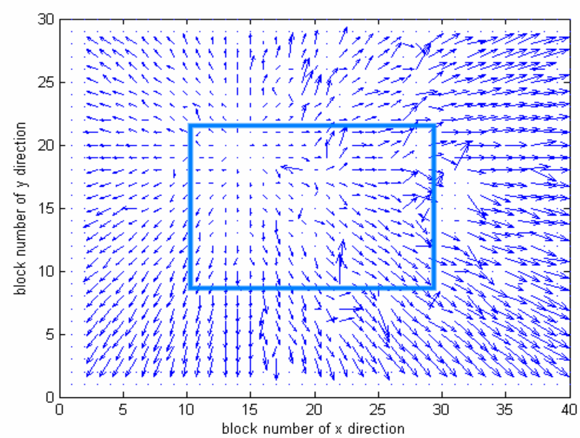


Figure 7. Motion field of zooming with camera movement.

The global motion induced by rapid camera panning has large value and high temporal correlation; therefore, we may use these properties to detect camera panning. The mean and variance of the global motion are computed over N samples as follows:

$$G_{mean}[n] = \frac{1}{N}\sum_{i=0}^{N-1} G[n-i] \quad \textbf{(5)}$$

$$G_{var}[n] = \frac{1}{N}\sum_{i=0}^{n-1} (G[n-i] - G_{mean}[n-i])^2 \quad \textbf{(6)}$$

where $G_{mean}[n]$ and $G_{var}[n]$, respectively, are the mean and variance of the global motion. Fig. 8 shows the accumulated global motion induced by rapid camera panning for test sequence 1 in our experiment. We choose N=7, $G_{mean}[n]>3$, and $G_{var}[n]<10$. Under these parameters, the rapid camera panning detected is shown by the red curve (marked with triangle) in Fig. 8.

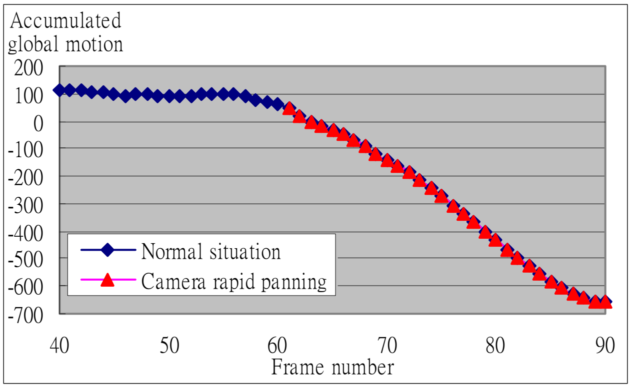


Figure 8. Detection of rapid camera panning.

IV. IMPACT OF DIGITAL IMAGE STABILIZATION ON CODING PERFORMANCE

A number of experiments were set up to analyze the impact of image stabilization on video encoding performance. We applied two different codecs running the simple profile (SP) of MPEG-4 [7] and the main profile (MP) of H.264 [8], both with constant quantization parameters (Qp's), on the test sequences. For MPEG-4, we used the Microsoft Reference Software and turned off all advanced coding features such as bi-directional VOP and global motion compensation. For H.264, we used the JM 8.1 reference software. The H.264 standard includes many new coding tools, such as multi-reference frames, quarter-pixel motion compensation, in-loop deblocking filter, motion vector prediction, and context-adaptive binary arithmetic coding, which are turned on in our experiments. The two encoders were configured at different performance levels. Table 1 lists the configurations of the two encoders in our experiments.

In the experiments, raw image sequences of various scenes were obtained using a camera and stored on a PC. Hand jiggles and optical zooming were introduced during the video capturing process in some cases. Each image sequence, before and after the image stabilization, was input to the two codecs to generate the compressed video. Note, however, the original image sequence was cropped to fit into the same size as the stabilized sequence before encoding. We report the coding results of two typical sequences here, Figs. 10-11. Both sequences are 300 frames long. The results of other sequences are very similar, and hence we omit them.

The numerical results are shown in Tables 2 and 3, where the averaged PSNR values and the total numbers of bits for motion vectors are shown. As indicated in the "Motion bit diff" column of Tables 2 and 3, the stabilized sequence uses a fewer number of bits for describing the motion vectors. This is due to the effect of image stabilizing, which smoothes the rapid motion fluctuation and reduces the chance for the motion vectors to go out of the motion search range. Therefore, the motion vectors of the stabilized sequence would have smaller values on the average. But the total bitrate for the stabilized sequence is slightly higher than that of the original sequence in most cases, as indicated in the "Bitrate diff" column of Tables 2 and 3.

While fewer bits are used to encode the motion information for the stabilized sequence, the increase in total bitrate indicates that more bits are spent on coding the texture data, mostly for the image blocks along the picture borders. The results also show that the increase in total bitrate may not necessarily result in a higher PSNR. In any case, the differences in PSNR and in total bitrate are insignificantly small, as we can see in Figs. 11-12.

TABLE I. 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
Reference frame(s)	1	5
Version	Microsoft	JM 8.1
Search range	[−16, +16]	
I frame period	Only one I frame	
Sequence I	304*224	
Sequence II	288*208	

TABLE II. PSNR AND BIT COUNTS FOR MOTION VECTORS OF SEQUENCE I CODED BY MPEG 4

Qp	Original		Stabilized		Motion bit diff	PSNR diff	Bitrate diff (%)
	PSNR	Motion bits	PSNR	Motion bits			
5	35.0021	830095	35.0078	823231	-6864	0.0057	2.92
10	30.1835	794271	30.1944	782001	-12270	0.0091	2.98
15	28.1763	791440	28.1911	771514	-19926	0.0148	2.71
20	26.8134	785251	26.824	765121	-20130	0.0106	1.52
25	25.9132	787658	25.9202	765869	-21789	0.0070	0.45
30	25.1570	786315	25.1606	766514	-19801	0.0036	-0.42

TABLE III. PSNR AND BIT COUNTS FOR MOTION VECTORS OF SEQUENCE II CODED BY MPEG 4

Qp	Original		Stabilized		Motion bit diff	PSNR diff	Bitrate diff (%)
	PSNR	Motion bits	PSNR	Motion bits			
5	36.3620	543569	36.3431	517133	-26436	-0.0189	0.88
10	32.9932	532702	32.9628	497875	-34827	-0.0304	0.17
15	31.5054	552357	31.4735	512583	-39774	-0.0319	0.22
20	30.4499	560585	30.4133	520769	-39816	-0.0366	0.29
25	29.7303	568956	29.6730	528078	-40878	-0.0573	0.30
30	29.1056	572619	29.0264	530171	-42448	-0.0792	0.46

Figure 9. Sequence I.

Figure 10. Sequence II.

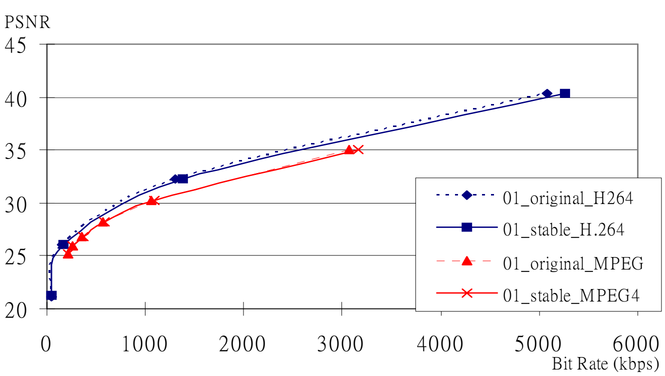


Figure 11. Comparison of the R-D curves of Sequence I.

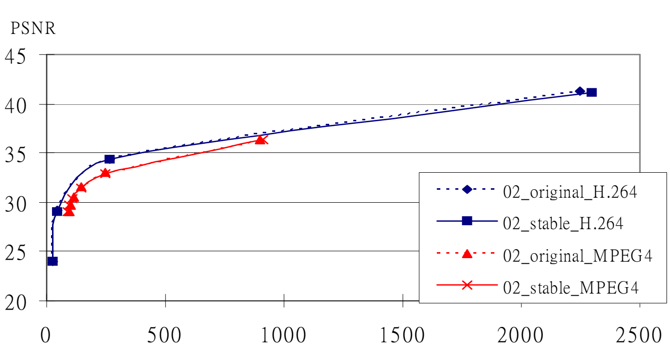


Figure 12. Comparison of the R-D curves of Sequence II.

V. Integration of Image Stabilizer with Video Encoder

The digital image stabilization algorithm described in the previous section requires the computation of a global motion vector for each video frame. Most previous algorithms consider the global motion estimation as a separate process [1]-[3]. Here, we take a different approach that reduces the total computational cost by integrating the digital image stabilizer with the video encoder. In [6], we have found that the local motion vector that happens most frequently in a frame can be considered as the global motion of the frame. Such motion vectors usually belong to the background region of the image. To reduce the computational cost, we confine the computation of the global motion to within the background region. Unlike previous approaches that use predefined regions to search for the global motion, we determine the background region dynamically for each frame. This allows us to improve the accuracy of global motion estimation.

The block diagram of the integration scheme is shown in Fig. 13, where the motion vectors computed by the video encoder are processed to detect the global motion. Note that the input to the video encoder is a sequence of stabilized images. Thus the detected global motion represents the image motion that is induced by camera panning. Then the background region is formed by macroblocks with motion vectors equal (or close) to the global motion. A smaller region inside this background region is used to compute the global motion required for stabilizing the next video frame. That is, the computation of the global motion for image stabilization is confined to within the selected background region. In our simulation, a 64*64 block is chosen to be the background prediction.

Therefore, the computation of motion estimation for the image stabilizer is reduced by a factor of (720*480)/(64*64)=84.375. Fig. 15 shows that the global motion in x and y direction estimated by the integration scheme is almost the same as the original image stabilizer described in [6]. The total absolute difference of global motion vectors over 300 frames between the integration scheme and the original scheme in the x direction is 22 pixels, so the average error is 0.073 pixels.

Note that the integration scheme only uses the motion information generated by the video encoder. The additional operation for detecting the background region does not affect the video encoder.

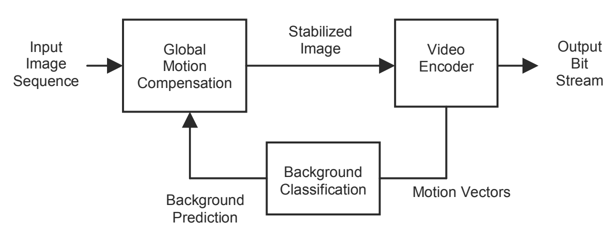


Figure 13. Integration scheme of the imahe stabilizer and the MPEG-4 encoder.

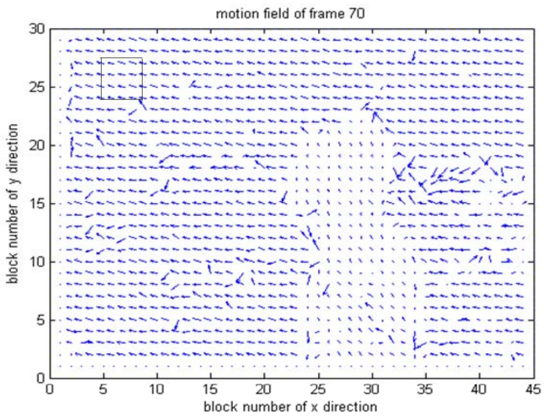
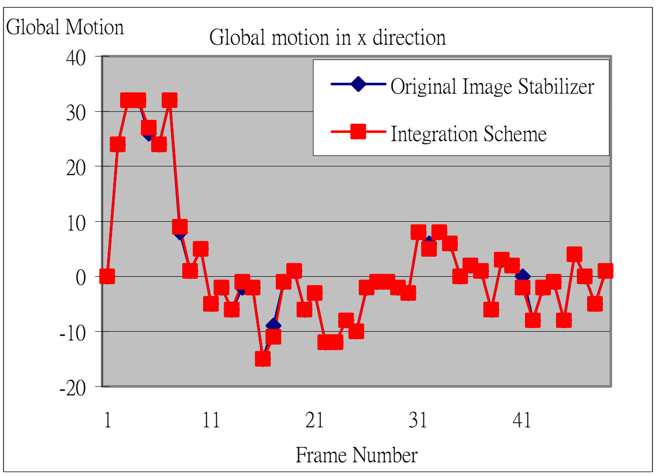
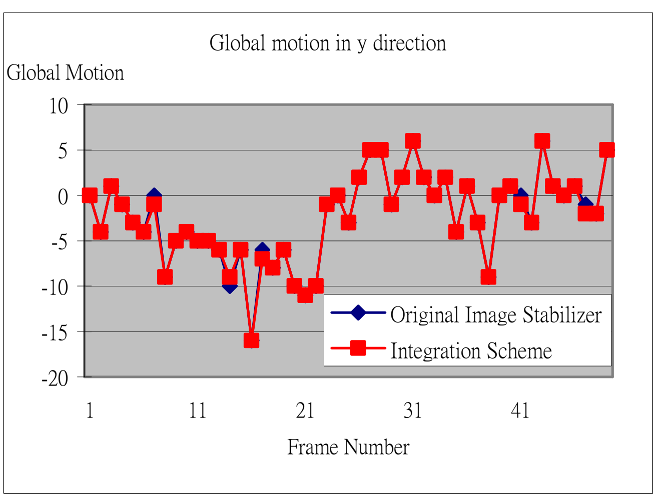


Figure 14. The predicted background region of frame 70.



(a)



(b)

Figure 15. The global motion estimated by the integration scheme against that by the original image stabilizer in (a) the x direction and (b) the y direction. Note that the two curves almost overlap.

VI. SUMMARY

In this paper, we have described an effective digital image stabilization system. For camera zooming, we compute the global motion over a block located at the center of the image and make use of the observation that the motion vectors contributed by hand jiggles are more dominant than those by optical zooming.

The effect of digital image stabilization on the video coding performance of MPEG-4 and H.264 is examined. It is found that more inter-coded blocks are resulted and the number of bits allocated for motion vectors are reduced. Unlike previous report [5] which uses synthetic jiggling, our experiments with real image sequences show that at the same PSNR level, digital image stabilization may not necessarily lead to total bit rate reduction. However, the difference is negligible. Nonetheless, digital image stabilization gives rise to significantly better perceptual quality.

To improve the total computational cost of a digital video camera, we have also investigated the integration of digital image stabilizer with video encoder. The integration scheme uses the motion vectors computed by the video encoder to select a 64x64 background block and confines the computation of global motion to this small region. Results obtained in the experiment show that the performance of the integration scheme is as good as that of the original image stabilizer described in [6].

REFERENCES

[1] K. Uomori, A. Morimura, H. Ishii, T. Sakaguchi, and Y. Kitamura, "Automatic image stabilizing system by full-digital signal processing," *IEEE Trans. Consumer Electron.*, vol. 36, no. 3, pp. 510-519, Aug. 1990.

[2] 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. Consumer Electron.*, vol. 37, no. 3, pp. 521-530, Aug. 1991.

[3] 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. Consumer Electron.*, vol. 38, no. 3, pp. 607-616, Aug. 1992.

[4] S. Erturk, "Image sequence stabilization by low-pass filtering of interframe motion," *Proc. SPIE, Visual Communication and image Processing*, vol. 4310, pp. 434-442, 2001.

[5] A. Engelsberg, and G. Schmidt, "A comparative review of digital image stabilising algorithms for mobile video communications," *IEEE Trans. Consumer Electron.*, vol. 45, Issue 3, pp. 591 – 597, Aug. 1999.

[6] C.-K. Liang, Y.-C. Peng, H.-A. Chang, and H. Chen, "The effect of digital image stabilization on coding performance", to be presented in *Int'l Symposium on Intelligent Multimedia, Video and Speech Processing*, Oct. 2004.

[7] *Generic Coding of Audio-Visual Objects – Part 2 Visual,* ISO/IEC 14496-2, 2001.

[8] *Draft ITU-T Recommendation H.264 and Final Draft International Standard of Joint Video Specification 14 496-10 Advanced Video Coding*, May *2003.*

[9] S. Erturk, and M. K. Gullu, "Membership function adaptive fuzzy filter for image sequence stabilization," *IEEE Trans. Consumer Electron.*, vol. 50, issue 1, pp.1-7, Feb. 2004.