THE EFFECT OF DIGITAL IMAGE STABILIZATION ON CODING PERFORMANCE

Chia-Kai Liang, Yu-Chun Peng, Hung-An Chang, Che-Chun Su, and Homer Chen

Graduate Institute of Communications Engineering, National Taiwan University
Email: {b89901103, b88901089, b89901049, b89901026}@ntu.edu.tw, homer@cc.ee.ntu.edu.tw

ABSTRACT

A digital image stabilization system compensates the image movement caused by hand jiggles and thereby improves the perceptual quality of the image sequence captured by a hand-held digital video camera. In this paper, we apply two recently developed video coding standards, MPEG-4 and H.264, to investigate the effect of image stabilization on video coding. The coding performance of these algorithms on the stabilized image sequence is compared to that on the original image sequence. The results show that image stabilization greatly improves the viewing experience and results in a decrease of bit counts for motion vectors, but the difference in the overall coding performance is minor. Details of the video coding experiments and the image stabilization system are described.

1. INTRODUCTION

Image sequence stabilization is an important requirement as video capture devices are being widely used in applications such as security surveillance, military reconnaissance, and consumer electronics. The goal of image sequence stabilization is to remove the involuntary image movements caused by, for example, hand jiggles or vibrations of the camera mounting devices.

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 in the temporal domain 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.

This paper is organized as follows. Section 2 describes the architecture of a digital image stabilization system we have developed. In section 3, we describe the experiments for testing the effect of digital image stabilization on the performance of two video encoders, MPEG-4 and H.264, followed by a conclusion.

2. SYSTEM ARCHITECTURE

Fig. 1 shows the architecture of the digital image stabilization system we have developed.

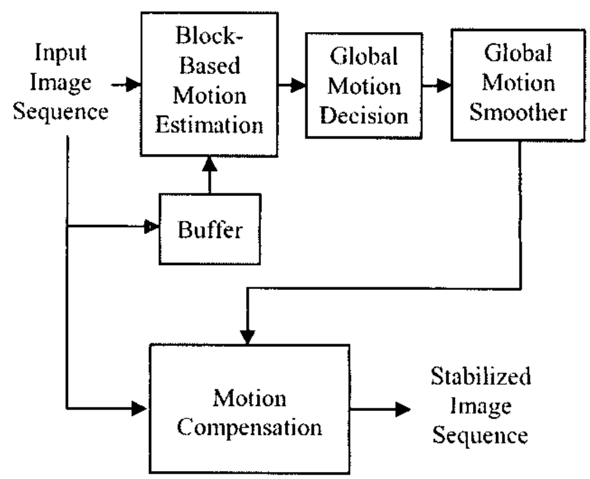

Figure 1. The system architecture of a digital image stabilizer.

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 decision 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.

2.1. Global Motion Decision

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.

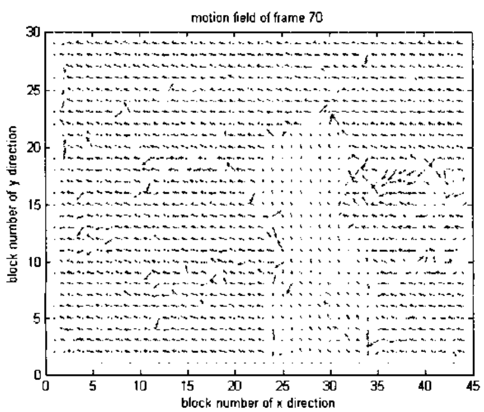

Figure 2. Motion field of frame 70.

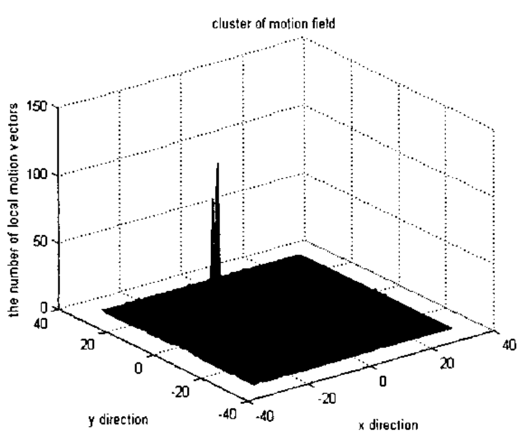

Figure 3. Clusters of a motion field.

2.2. 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

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

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

For computational efficiency, we use a causal lowpass filter that would not cause frame delay. 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.

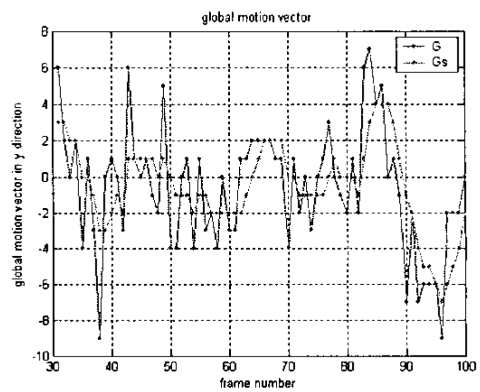

Figure 4. The global motion before and after smoothing.

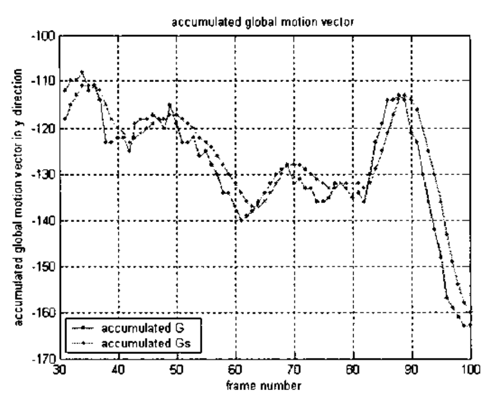

Figure 5. Accumulated global motion before and after smoothing.

3. 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 [6], 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, in some cases, optical zooming were introduced during the video capturing process. 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. 6-7. 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 and3, 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 or may not result in a better PSNR. In any case, the differences in PSNR and in total bitrate are insignificantly small, as we see in Figs. 8-9 that the R-D curves generated by each encoder for the stabilized image sequence and the original sequence are too close to call for a winner.

Table 1. 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 2. 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 3. 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

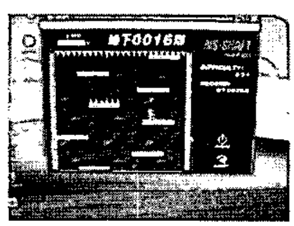

Figure 6. Sequence I.

Figure 7. Sequence II.

5. CONCLUSION

We have described an image stabilization system and examined its effects on the video coding performance of MPEG-4 and H.264. We have found that image stabilization results in more inter-coded blocks and reduces the number of bits allocated for motion vectors. Unlike previous report [5] which uses synthetic jiggling, our experiments with real image sequences show that image stabilization may not necessarily lead to a reduction in the total bit rate. In addition, the resulting PSNR of the encoded sequence remains almost the same. Overall, the changes in PSNR and total bitrate are negligible, despite that the perceptual quality of the stabilized image sequence is significantly improved.

It should be noted that when we first started out the experiments, a DV camera was used for video capture. The DV camera has a built-in motion JPEG compressor. So the image sequence input to the stabilizer represents a decompressed video clip. The image stabilizer described in this paper has no problem dealing with decompressed video. However, the decompressed video does have a dramatic impact on the coding performance of the MPEG-4 and H.264 encoders and obscure the effect of image stabilization. In this case, we have found that the difference in PSNR widens as the bitrate increases and that the original image sequence always outperforms the stabilized image sequence. It turns out that this faulty result is due to the fact that block re-decomposition, which happens when encoding the stabilized images, gives rise to higher residual error as compared to that of the original images and consequently results in inferior coding performance.

In the current implementation, the motion estimator of the image stabilization system is separated from the one in the video codec. To be cost-effective, these two modules have to be integrated. Further research towards this goal is underway.

4. ACKNOWLEDGEMENT

This work was supported in part by the National Science Council of Taiwan under contract NSC93-2752-E-002-006-PAE and by Quanta Computer Inc. under contract 93C010.

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 stabilisation 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, no. 3, pp. 591 – 597, Aug. 1999.

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

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

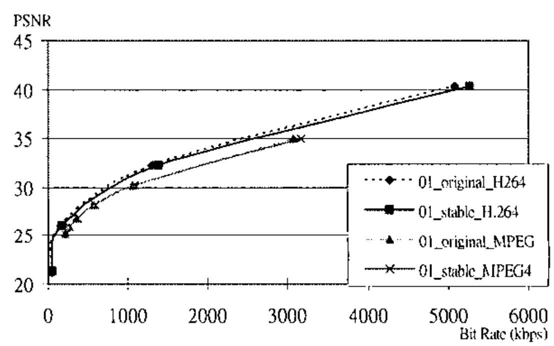

Figure 8. Comparison of the R-D curves of Sequence I.

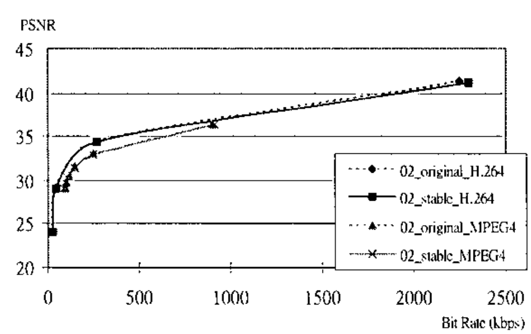

Figure 9. Comparison of the R-D curves of Sequence II.