

DSP IMPLEMENTATION OF DIGITAL IMAGE STABILIZER

Yu-Chun Peng, Meng-Ting Lu and Homer Chen

Graduate Institute of Communications Engineering, National Taiwan University

Email: {b88901089, b88901124}@ntu.edu.tw, homer@cc.ee.ntu.edu.tw

ABSTRACT

A digital image stabilization system compensates the image movement caused by hand jiggle for the image sequence captured by a hand-held video camera. In this paper, a simplified stabilization algorithm based on our previous work is presented. The algorithm performs block-based motion estimation on 16 local 16x16 blocks and uses a median filter to estimate the global motion. It reduces the complexity by confining the motion estimation to a small number of blocks of the image. This greatly facilitates the implementation of the algorithm on BF561, a DSP processor of Analog Device. Details of the DSP implementation are described.

1. INTRODUCTION

Image sequence stabilization is becoming an important requirement as video capture devices are extensively used in applications such as security surveillance, military reconnaissance, and consumer electronics. The task of image sequence stabilization is to remove the involuntary image movement caused by, for example, unstable hand-shaking or vibration.

Although traditional mechanical and optical devices have been used, digital video processing approaches are preferred in many applications for which cost and compactness of implementation are a major concern. The digital image stabilization problem has been extensively 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 inter-frame motion. A comparative review of the stabilization for mobile video communications is presented in [5].

For videos captured by digital video cameras, the digital image stabilizer computes the image motion and compensates for the inter-frame global motion induced by

hand motion. For this purpose, block-based motion estimation generally delivers satisfactory performance. For example, in [6] we describe an image stabilization algorithm that computes the block motion vector for the entire frame and estimates the global motion by clustering. However, the block-based motion estimation is a computationally intensive operation and becomes a bottleneck for some real-time applications of the digital stabilizer. In this paper, a simplified digital image stabilization algorithm is described that confines the motion estimation to 16 blocks. In addition, a median filter is applied to estimate the global motion. The simplification greatly facilitates the implementation of the algorithm on a DSP platform with BF561 as the core.

This paper is organized as follows. The architecture of the digital image stabilizer is described in Section 2. We introduce our DSP platform and the implementation work in Section 3, followed by a conclusion in Section 4.

2. DIGITAL IMAGE STABILIZATION ALGORITHM

For videos captured by digital video cameras, hand jiggle or camera panning introduces a global motion between successive frames and accounts for the motion of the background region in the image. Typically, 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.

The architecture of the image stabilizer is shown in Fig. 1. First, the 16 local motion vectors are computed by

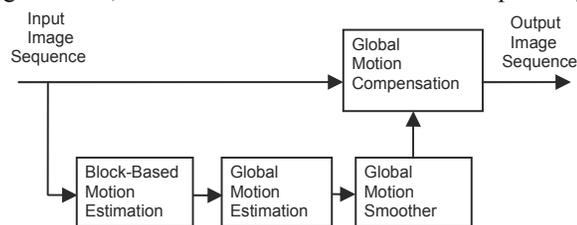


Figure 1. The block diagram of the image stabilizer.

block-based motion estimation. The resulting motion vectors are input to the global motion decision module to determine the global motion by applying a median filter. 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 estimation

As mentioned, the main task of the motion estimation in the digital image stabilizer is to compute the local motion vectors, which in turn are used to estimate the global motion of the image. The computation required for block-based motion estimation is a major burden for the image stabilizer in some applications. In order to save the computation, the block-based motion estimation is only performed on a number of selected background regions instead of the whole image. We choose 16 16x16 blocks around the image as the pre-located background region, which are shown in Fig. 2.

Thus the number of block motion vectors is reduced to 16. However, some of these local motion vectors could be unreliable. Thus a median filter is adopted into the global motion estimation process to overcome the effect of unreliable local motion vectors. The median filter is shown in Eq. (1).

$$G[n] = \text{median} \{MV_i, i=1, 2, \dots, 16\}, \quad (1)$$

where $G[n]$ denotes the estimated global motion vector for frame n . Fig. 3 and 4 show the global motion of two test sequences estimated by our simplified image stabilization algorithm. The results are very close to that estimated by the original image stabilizer [6]. The total absolute difference of global motion vectors over 300 frames between the simplified algorithm and the original algorithm [6] is 70 pixels, so the average error is 0.233 pixels.

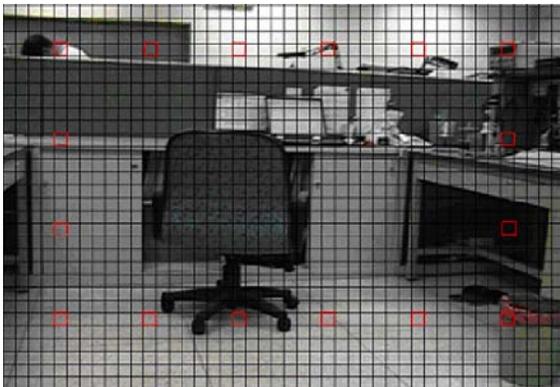


Figure 2. Pre-located background blocks

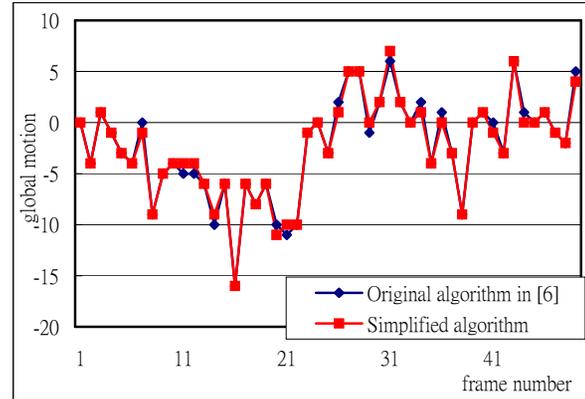


Figure 3. Comparison of global motions estimated by original and simplified algorithm for sequence 1.

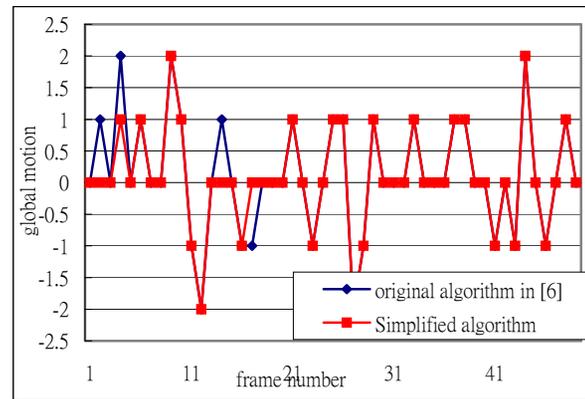


Figure 4. Comparison of global motions estimated by original and simplified algorithm for sequence 2.

2.2. Global motion smoother

We apply a lowpass filter similar to that in [4] to smooth out 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

$$G_s[n] = \sum_{i=0}^m a_i * G[n-i] \quad (2)$$

$$AG_s[n] = AG_s[n-1] + G_s[n], \quad (3)$$

$G[n]$ 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.

For computational efficiency, we use a causal lowpass filter that would not cause frame delay. Eq. (2) can be considered, as a weighted moving average filter with window size m . G_s gets smoother as m increases. Fig. 5 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. 6. As we can see, the stabilization effect is achieved.

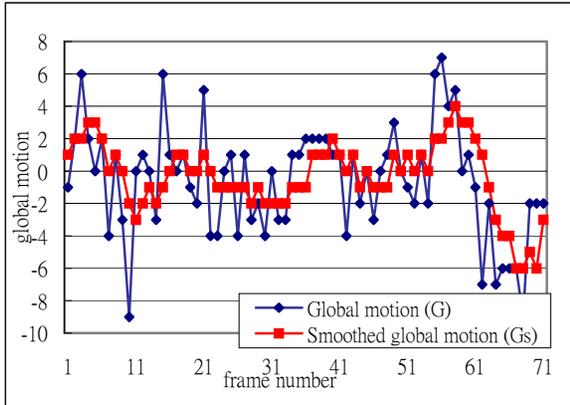


Figure 5. The global motion before and after smoothing.

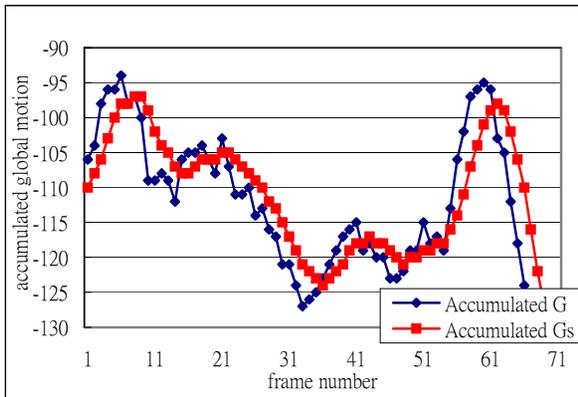


Figure 6. Accumulated global motion before and after smoothing.

3. DSP IMPLEMENTATION

We adopted ADSP-BF561 EZ-KIT Lite as our platform to implement the digital image stabilizer. ADSP-BF561 is a very powerful dual-core DSP processor developed by Analog Devices, Inc. At the heart of this device are two independent Analog Device Blackfin processors operating at 750 MHz. It is very suitable for real-time video signal processing applications because it has two Parallel Peripheral Interface (PPI) ports which allow our system to input and output video signal at the same time. The input PPI port is connected to ADV7183A advanced 10-bit video decoder, and the output PPI port is connected to ADV7179 chip scale NTSC/PAL video encoder. With these two chips, we can easily implement a CCIR 656 NTSC compatible system. ADSP-BF561 also has 128 KB on-chip L2 SRAM and 64 MB external SDRAM which is quite sufficient for performing motion estimation. Besides, two Direct

Memory Access (DMA) control modules are in charge of moving data in our system without adding any burden to the processor. The block diagram of the modules we exploited in our implementation is illustrated in Fig. 7.

The software architecture of our implementation is illustrated in Fig. 8. We set up three buffers at the input port and two buffers at the output port. Three buffers at the input port are necessary because we need one buffer for reference frame, one buffer for current frame, and one buffer for input data. At the output port, no reference frame is needed, so we can just use two buffers. In our system, descriptor-based DMA control is used for both input and output. When motion stabilization is started, the output port is blocked until the input port finishes filling in two input buffers and copy the second input buffer to the first output buffer. Then, our system modifies the DMA descriptors to lock input port at the third buffer, and output port at the first buffer. At the same time, taking the first input buffer as reference frame and the second input buffer as current frame, the image stabilization process is equally split for two processors. After finishing the image stabilization task and writing the stabilized data to the second output buffer. Our system changes the DMA descriptors to lock input port at the first buffer and output port at the second buffer. At the same time, taking the second input buffer as reference frame and third input buffer as current frame, the stabilized image is computed and written to the first output buffer. The same process continues until the application shuts down.

Table I. DSP implementation performance

	Processing Cycles	Output Frame Rate	Memory Copy
Non-optimized	460M	1.63 fps	5.76 fps
MDMA-optimized	250M	3 fps	8 fps

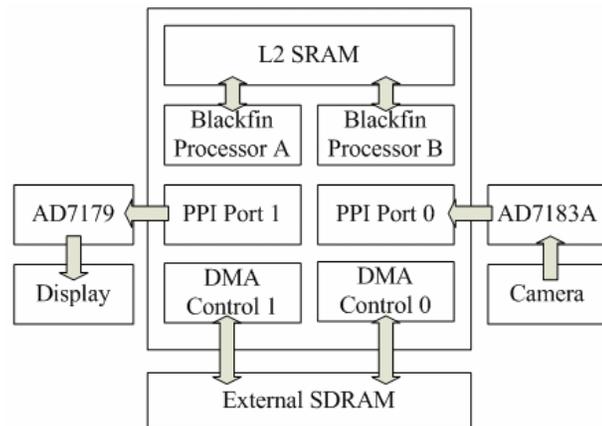


Figure 7. DSP block diagram.

Based on the above architecture, we have successfully implemented an experimental platform on ADSP-BF561. The video sequence captured by camera is processed to achieve stabilization and displayed with NTSC format. The performance of our non-optimized and Memory DMA (MDMA) optimized system is listed in table I. The second column is the cycles needed to do image stabilization for each frame. The third column is the frame rate of our system. The fourth column is the frame rate achieved by only doing memory copy without any processing. We can see from this table that the performance is primarily limited by the memory copy speed. Because our system is mainly written in C++, the performance should be further improved if we replace the key parts by assembly code and change the memory copy mechanism.

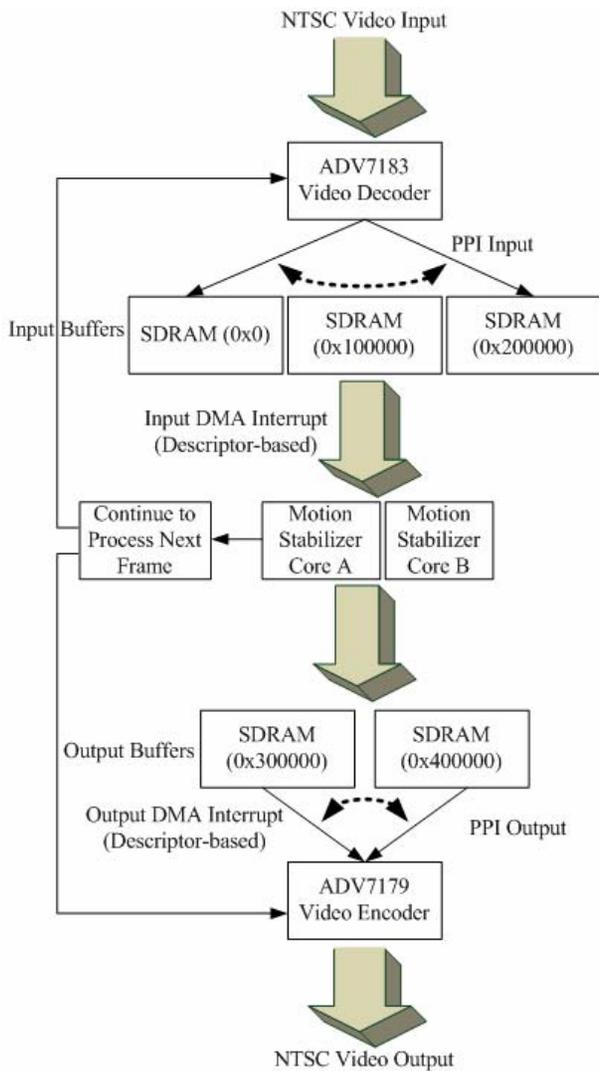


Figure 8. Software architecture.

4. ACKNOWLEDGEMENT

This work was supported in part by the National Science Council of Taiwan under contract NSC 93-2213-E-002-137, NSC 93-2219-E-002-031 and NSC93-2752-E-002-006-PAE.

5. CONCLUSION

We have presented an image stabilization algorithm that reduces the system complexity by performing block-based motion estimation on 16 local 16x16 blocks and using a median filter to estimate the global motion. The proposed system is implemented on a DSP-based platform, ADSP-BF561 EZ-KIT Lite. The results show that the system has low computational cost and is efficient for real-time applications such as digital video camera and digital camera module on mobile devices.

5. 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] C.-K. Liang, Y.-C. Peng, H.-A. Chang, and H. Chen, "The effect of digital image stabilization on coding performance," *Proc. IEEE, Int'l Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 402-405, 2004.
- [7] Y.-C. Peng, H.-A. Chang, Homer H. Chen, and C.-J. Kao, "Digital Image Stabilization and Its Integration with Video Encoder," *Proc. IEEE, Consumer Communications and Networking Conference*, 2005.
- [8] *ADSP-BF561 Blackfin Processor Hardware Reference Preliminary Revision 0.2*, Analog Devices Inc., Nov. 2003.