Digital camcorder image stabilizer based on gray-coded bit-plane block matching

Yeou-Min Yeh
National Chiao Tung University
Institute of Electronics Engineering
Hsinchu, 30010, Taiwan, R.O.C

Huang-Cheng Chiang
Industrial Technology Research Institute
Hsinchu, 30010, Taiwan, R.O.C

Sheng-Jyh Wang
National Chiao Tung University
Institute of Electronics Engineering
Hsinchu, 30010, Taiwan, R.O.C

Abstract. We propose an efficient algorithm to eliminate the nonpleasing effect caused by involuntary hand movement of camera holders. In our approach, 1-bit gray-coded bit-plane block matching, instead of 8-bit gray-level block matching, is used to greatly simplify the computation of motion estimation. This computation saving makes possible a finer division of image frame and thus facilitates the employment of a much more robust procedure for motion decision. To deal with various interfering factors in motion estimation, the temporal information of each local motion vector is also used to efficiently distinguish random-like movement from temporally correlated movement. To compensate for camera rotation, an affine model is used in the motion compensation unit without adding too much computation load. Having considered both programming flexibility and hardware efficiency, the motion decision unit and the motion compensation unit are coded in a microprocessor that interconnects with the stabilization hardware, which consists of the motion estimation unit and the digital zooming unit. A slightly simplified version of the proposed stabilizer is implemented on a field programmable gate array (FPGA) board.
[DOI: 10.1117/1.1405415]

Subject terms: digital image stabilization; motion estimation; digital camcorder; gray-coded bit-plane.

Paper PT-002 received Nov. 30, 2000; revised manuscript received Feb. 10, 2001; accepted for publication Feb. 21, 2001.

1 Introduction

Recently, more and more video cameras include compact size and powerful zooming capability. The advancement of these features makes the image stability problem even more crucial, because an unconscious movement of the holding hand may cause a dramatic shaking of the images. As a consequence, an image stabilization system is usually required to relieve the problem. Among various types of stabilization systems, a digital image stabilization (DIS) system, which can be fully realized in very large scale integration (VLSI), could be a more appropriate solution to fit the compactness requirement. So far, many approaches regarding DIS have been proposed and some of them have already been implemented in commercial video cameras.

Figure 1 shows a typical structure of a digital video camera, equipped with a DIS system and a corresponding frame memory[1] (FM). The FM is used to store current image data and to output the stabilized image data. In general, as shown in Fig. 2, a DIS system usually includes five major units: (1) preprocessing unit, (2) a motion estimation unit, (3) a motion decision unit, (4) a motion compensation unit for FM, and (5) a digital zooming unit.[2]

A traditional way to do motion estimation is to use block-matching methods.[1–7] To reduce computational complexity, these block-matching methods usually divide an image into a small number of blocks and select some representative points to calculate the motion vector of each block. Then, these block motion vectors are utilized together to estimate the global motion vector to compensate the movement of the whole image. Even though this coarse-division strategy can save a huge amount of computation, the rough division of an image may cause the loss of local information and reduce the precision in global motion decision. Without decreasing too much accuracy in motion estimation, Lee et al. proposed in Refs. 8 and 9, respectively, the usage of bit-plane and gray-coded bit-plane to do block matching. Even though this approach has greatly reduced the computation complexity, their algorithms are still based on some traditional methods for block division, motion decision, and motion compensation. For these conventional methods, only simple strategies can be applied in motion decision and the resulting motion compensation is not very reliable.

In this paper, we also adopt the gray-coded bit-plane strategy to do motion estimation. Under an acceptable increase of computation complexity, however, we divide an image frame into finer blocks to do localized block matching. This finer division enables a much more accurate estimation of the local movement inside the captured frames. Moreover, the increased number of local motion vectors (LMVs) facilitates the employment of a more complex procedure for motion decision. Furthermore, to distinguish the movement caused by camera shaking from the movement caused by moving objects or intentional panning, the tem-

This paper is a revision of a paper presented at the SPIE conference on Input/Output and Imaging Technologies II, July 2000, Taipei, Taiwan. The paper presented there appears (unrefereed) in *SPIE Proceedings* Vol. 4080.

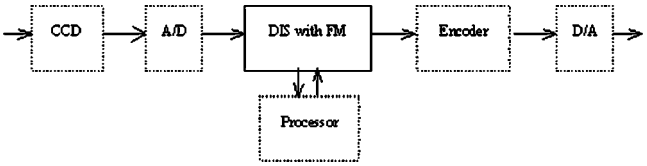


Fig. 1 Block diagram of a digital video camera with a DIS system.

poral correlation of each local motion vector is also carefully investigated in our DIS system. In motion compensation, an affine model is evaluated to compensate translational and rotatory movements. Beside the software development of the proposed stabilization algorithm, we also implement the motion estimation part in an efficient real-time hardware on a field programmable gate array.

2 Localized Block Matching Over the Gray-Coded Bit-Plane

Traditionally, operations are applied directly on 8-bit gray-level images to do image stabilization. The involved 8-bit operations, especially the 8-bit block matching in motion estimation, result in a heavy computational load for real-time hardware implementation. In 1998, Ko et al. proposed the usage of bit-planes in image stabilization.[8] With bit-planes, the block matching process can be implemented using only binary Boolean operations and the computation complexity of motion estimation can be significantly reduced without sacrificing too much motion estimation performance. In 1999, Lee et al. proposed the usage of gray-coded bit-planes to further improve the accuracy of local motion vectors.[9] In this paper, we also adopt gray-coded bit-planes as the basis of motion estimation. Here, assume $f(x,y)$ is a gray-level image and is represented as:

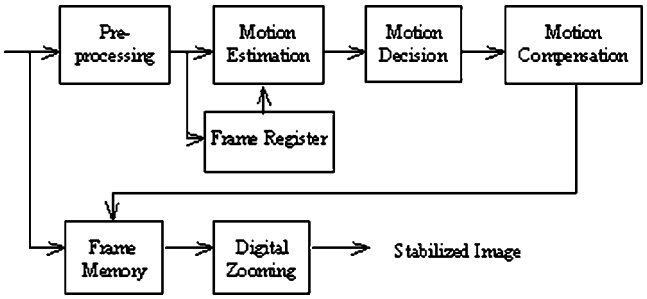


Fig. 2 General structure of DIS system with FM.

$$f(x,y)=a_{K-1}(x,y)2^{K-1}+a_{K-2}(x,y)2^{K-2}+\cdots +a_1(x,y)2+a_0(x,y). \quad (1)$$

Then, the gray-coded bit-planes are defined as:

$$g_i(x,y)=a_i(x,y)\oplus a_{i+1}(x,y), \quad 0\leqslant i\leqslant K-2, \quad (2)$$

and

$$g_{K-1}(x,y)=a_{K-1}(x,y).$$

Figure 3 shows the comparison between bit-planes and gray-coded bit-planes. We can easily see that either the fifth to seventh bit-planes or the fifth to seventh gray-coded bit-planes can roughly catch the spirit of image contents, but the gray-coded bit-planes tend to have less intensity fluctuation. In this paper, we work on the fifth or the sixth gray-coded bit-planes to do motion estimation and the involved correlation measure is defined as:

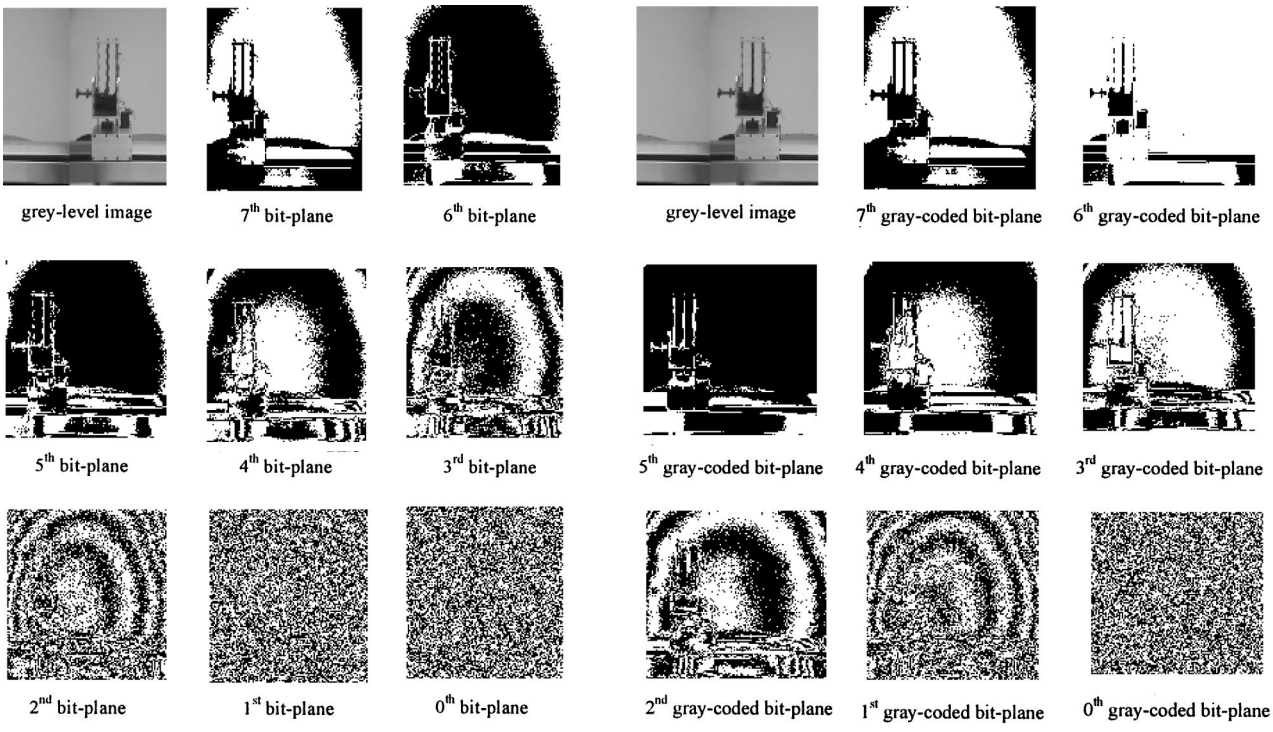


Fig. 3 Bit-planes versus gray-coded bit-planes.

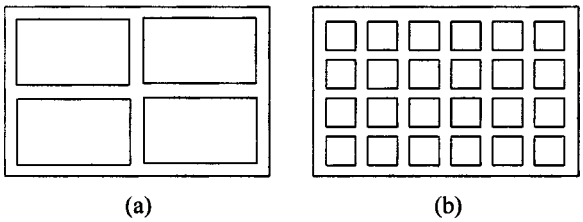


Fig. 4 (a) Coarse division versus (b) fine division.

$$c(m,n)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} g_k^t(x,y)\oplus g_k^{t-1}(x+m,y+n). \quad (3)$$

To improve the motion estimation accuracy, we apply a finer division over the gray-coded bit-planes. In Fig. 4, we demonstrate the comparison of a traditional rough-division method versus our fine-division method. Since the operations over gray-coded bit-planes are much simpler than the operations over 8-bit gray-level images, the computation complexity of our fine-division 1-bit gray-coded bit-plane approach is roughly the same as the traditional coarse-division 8-bit gray-level image approach.

The image in Fig. 5 is a frame extracted from an image sequence, which is captured by an intentionally shaken video camera. The scene in this image sequence contains a moving object, which is moving to the right. The traditional coarse division method divides this frame into four blocks and detects four local motion vectors separately: (4,0), $(-1,-6)$, $(-5,1)$, and $(-2,2)$ [see Fig. 5(a)]. Three of these four LMVs, except the lower-right LMV, are not reliable due to either lack of features or the presence of repeated patterns. The lower-right LMV, on the other hand, is also biased due to the existence of a moving object within that block. Therefore, for this image sequence, we could hardly detect the actual movement caused by the shaking video camera. As a comparison, with the proposed fine-division approach, plus some motion decision strategy that is mentioned later, there are many LMVs that may still indicate the movement of the video camera [as shown in Fig. 5(b)].

With the fine-division approach, the presence of some moving objects in the image frame will have less impact on the accuracy of global motion estimation. Moreover, the increased amount of motion vectors also increases the robustness of motion decision. Regarding the choice of block size, it is actually a trade-off issue. If the block size is too small, the accuracy of motion estimation is decreased; while if the block size is too large, some local information will get lost. Consequently, for a practical camera system, we empirically choose the block size to be 64×64 and divide each frame into 24 regions, as shown in Fig. 6. Moreover, since the core operation in block matching is actually the XOR operation on binary images, we adopt the full search approach with the search range being ±14.

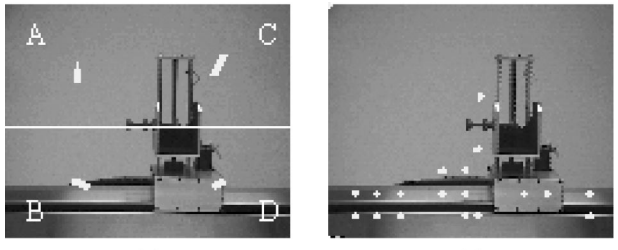


Fig. 5 (a) Coarse division of image and the detected LMVs and (b) fine division of image and the valid LMVs.

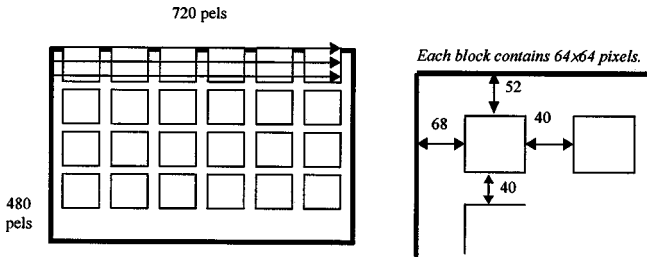


Fig. 6 Illustration of the division of the gray-coded bit-planes in our DIS system.

3 Motion Decision

Many factors exist that can affect the accuracy and performance of motion estimation, which we call ''interfering factors.'' Among these factors, lack of features, existence of repeated patterns, existence of moving objects, intentional panning, intentional zooming, and low SNR are most common. Many methods for detecting these factors have already been proposed.[1,3,4,10] However, these methods may not be suitable for our architecture and we design our own approach to detect these factors for localized block matching over gray-coded bit-planes.

3.1 *Lack of Features*

Lack of feature in a block means the image content in that block does not have enough features to characterize motion. In this case, the estimated LMV for that block is not reliable and should not be used for global motion compensation. To detect the occurrence of lack of feature, we check the correlation value $c(m,n)$, which has been defined in Eq. (3) for each block. The average correlation C_{ave} and the minimum correlation $C_{\min}$ are defined as:

$$C_{\text{ave}}=\frac{1}{(2P+1)(2Q+1)}\sum_{m=-P}^{P}\sum_{n=-Q}^{Q} c(m,n), \quad (4)$$

and

$$C_{\min}=\min_{m,n} c(m,n), \quad (5)$$

where $m\in[-P,P]$ and $n\in[-Q,Q]$ are the indices of the search neighborhood in the horizontal and vertical directions. For a lack-of-feature block, the correlation values for different (m,n) would be very similar. Hence, as the difference between C_{ave} and $C_{\min}$ is smaller than a predefined threshold, we declare that this block is lack of features and the estimated LMV is invalid.

3.2 *Existence of Repeated Patterns*

If a block contains repeated patterns, the similarity of image content due to the repeated patterns may cause a mis-

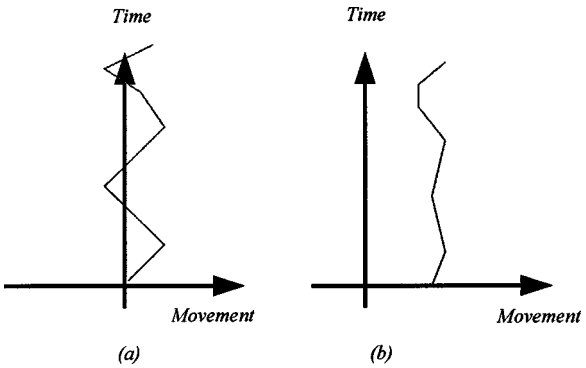


Fig. 7 Two kinds of motion: (a) random-like motion and (b) temporally correlated motion.

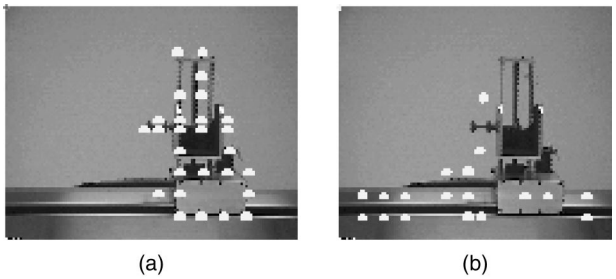


Fig. 8 Simulation results: (a) temporally correlated LMVs and (b) random-like LMVs.

judgment of motion estimation. This problem becomes more serious when the block matching process is localized. To detect the occurrence of repeated patterns, we also check the correlation value $c(m,n)$ in block matching. If there is a repeated pattern, the first minimal correlation value $C_{\text{1st-min}}$ and the second minimal correlation value $C_{\text{2nd-min}}$ are usually very similar. A simple thresholding mechanism can thus be applied on each block to detect the existence of repeated patterns.

3.3 *Existence of Moving Objects*

If an image sequence contains a moving object, the regions including this moving object may offer incorrect LMVs. We must eliminate these invalid LMVs to ensure the accuracy of motion compensation. Here, we propose a new approach, which is efficient and can be easily implemented for the detection of moving objects. First, we present the difference between two major types of motion: random-like motion and temporally correlated motion. As shown in Fig. 7(a), a motion regarded as random-like will fluctuate around zero and the variance of this motion is usually large. On the other hand, Fig. 7(b) shows a temporally correlated motion, which usually moves in a specific direction and the variance of this motion is usually small.

These two types of motion are closely related to the motion caused by hand shaking and the motion caused by moving objects or intentional panning. The motion caused by hand shaking causes the captured scene to fluctuate around the center of focus, which causes the motion vectors to fluctuate around zero. On the other hand, the movement caused by moving objects or intentional motion tends to move in the same direction for a short time. Consequently, we can classify the motion caused by hand shaking as random-like motion and the motion caused by moving objects or intentional panning as temporally correlated motion. Based on this observation, we design a simple test, as shown next, to distinguish these two kinds of motion:

$$|\mathrm{LMV}(t_1)-\mathrm{LMV}(t_2)|+|\mathrm{LMV}(t_2)-\mathrm{LMV}(t_3)|+\cdots + |\mathrm{LMV}(t_{N-1})-\mathrm{LMV}(t_N)|=T_1, \tag{6}$$

$$\frac{1}{N}\sum_{i=1}^{N}\mathrm{LMV}(t_i)=T_2. \tag{7}$$

If $T_1/T_2<K_1$ and $T_2>K_2$, then it is temporally correlated motion; otherwise it is random-like motion.

Given a block, we observe its LMV along the temporal domain. Assume $\mathrm{LMV}(t_i)$ denotes the LMV at time t_i. If a motion behaves as temporally correlated motion, T_1 is usually small and T_2 is usually large. After these temporally correlated motion vectors are detected, we use them as clues to detect moving objects. In our simulation, we choose $N=8$, $K_1=5$, and $K_2=1$. Figure 8 shows the experiment result. The test sequence contains two motions: temporally correlated motion at the slider [Fig. 8(a)] and random-like motion for the remaining parts [Fig. 8(b)]. The simulation result demonstrates that these LMVs corresponding to temporally correlated motion are correctly detected and they are gathering around the slider [as shown in Fig. 8(a)]. On the other hand, Fig. 8(b) indicates these random-like LMVs. Note that in this example we have already applied the lack-of-feature test to remove some unreliable LMVs.

If the temporally correlated motion vectors appear to be localized, these motion vectors are treated as being caused by existing moving objects. On the other hand, if the temporally correlated motion vectors have a global trend, the camera may be under an intentional panning. This situation is discussed next.

3.4 *Intentional Panning*

The intentional motion of camcorders, like the motion of panning, may cause a misjudgment of the motion compensation. Thus, reliable detection of a panning condition is required in the motion decision unit. As mentioned, temporally correlated movement behaves differently from random-like movement, and we can categorize the motion of panning as temporally correlated. Moreover, even though both intentional panning and the existence of moving objects will cause temporally correlated movement, the occurrence of panning will have a global influence, while the existence of moving objects tends to have a local influence. Hence, in our approach, if more than 80% of the LMVs are detected as temporally correlated, we consider that the camera is under a panning condition and no motion compensation is required. Otherwise, we assume these temporally correlated motion vectors are caused by some moving objects in the image and declare these LMVs as invalid.

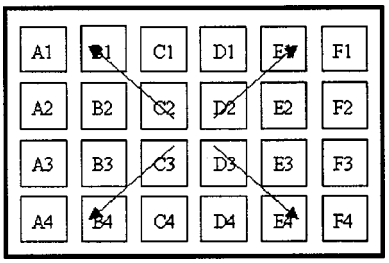


Fig. 9 Division of image frame and the indicated directions of motion vectors in the zoom-in condition.

3.5 *Intentional Zooming*

One of the popular functions in digital camcorders is to optically zoom in or zoom out of the scene. When this situation occurs, the motion decision unit should detect it and avoid the misusage of motion compensation. In our DIS system, the local motion vectors are allocated as shown in Fig. 9. To detect intentional zooming, we can check the LMVs at B1, C2, D2, E1, C3, B4, D3, and E4. If these LMVs are detected to have motion vectors pointing along the directions as indicated in Fig. 9, we judge that the camera is under the zoom-in condition and no motion compensation is required. In a similar way, we can also detect the occurrence of the zoom-out condition.

3.6 *Low SNR*

If the SNR of the image content is too small, the accuracy and performance of motion estimation will be affected. Besides, some image content, such as a waving sea, may introduce an unstable effect on motion estimation. Therefore, we design a noise-level test to check whether the remaining valid LMVs are similar enough to each other. If yes, these valid LMVs could be used to compensate the global movement caused by the vibration of video camera; otherwise, these LMVs could be too disordered to be used.

To check the noise level, we compute the variance of these valid LMVs. If the variance is higher than a predefined threshold, we regard these LMVs as useless and no motion compensation is made. Moreover, if the number of these valid LMVs is too small, these LMVs might not provide accurate enough information for motion compensation. When this happens, we also disable the use of motion compensation.

Figure 10 shows the procedure adopted in our motion decision unit. Note that the feasibility of this procedure actually comes from the fine-division strategy. Without fine division, the number of LMVs will not be large enough to support these tests. After this motion-decision procedure, we output these valid LMVs, together with the inferred status, to the motion compensation unit.

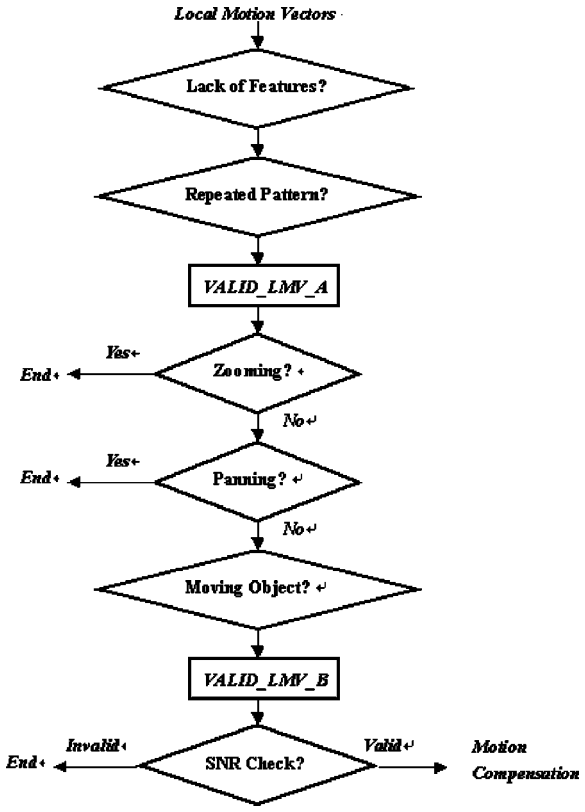


Fig. 10 Motion decision procedure.

4 Motion Compensation

After LMVs are detected and verified, these valid LMVs are combined together to estimate the global motion vector for motion compensation. Since the shaking of the camera usually consists of translational movement and rotatory movement, the affine model is used in this paper to describe the movement of the global motion. Equation (8) shows the equations of affine motion.

$$\begin{cases} \overline{X}_{t+1} = aX_t + bY_t + c \\ \overline{Y}_{t+1} = dX_t + eY_t + f \end{cases}, \tag{8}$$

where $(\overline{X},\overline{Y})$ denote the coordinates of the compared frame and (X,Y) denote the coordinates of the reference frame.

To estimate the six coefficients (a to f) in the affine model, we use the least mean squares approach. Assume there are N valid motion vectors. We use the standard optimization method to find the ''optimal'' coefficients that minimize the following equations:

$$\sum_{n=1}^{N} (aX_n + bY_n + c - \overline{X_n})^2, \qquad \sum_{n=1}^{N} (dX_n + eY_n + f - \overline{Y_n})^2. \tag{9}$$

After some straightforward mathematical deductions, these six coefficients can be calculated by

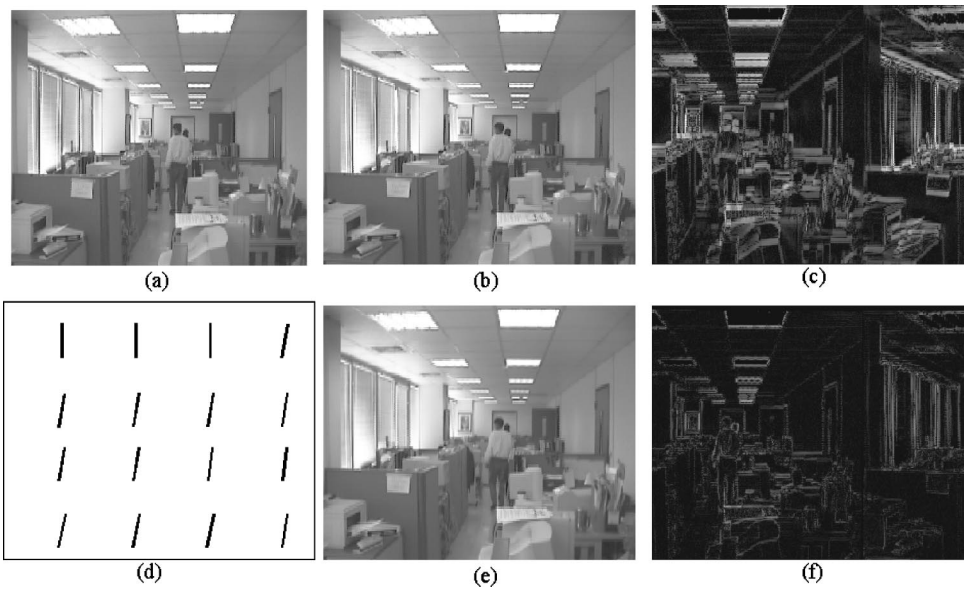


Fig. 11 (a) Reference frame, (b) current frame, (c) difference between (a) and (b), (d) LMVs, (e) frame after compensation, and (f) difference between (a) and (e).

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} X_1^2+X_2^2+\cdots+X_n^2 & X_1Y_1+X_2Y_2+\cdots+X_nY & X_1+X_2+\cdots+X_n \\ X_1Y_1+X_2Y_2+\cdots+X_nY_n & Y_1^2+Y_2^2+\cdots+Y_n^2 & Y_1+Y_2+\cdots+Y_n \\ X_1+X_2+\cdots+X_n & Y_1+Y_2+\cdots+Y_n & n \end{bmatrix}^{-1} \times \begin{bmatrix} \overline{X_1}X_1+\overline{X_2}X_2+\cdots+\overline{X_n}X_n \\ \overline{X_1}Y_1+\overline{X_2}Y_2+\cdots+\overline{X_n}Y_n \\ \overline{X_1}+\overline{X_2}+\cdots+\overline{X_n} \end{bmatrix},$$

$$\begin{bmatrix} d \\ e \\ f \end{bmatrix} = \begin{bmatrix} X_1^2+X_2^2+\cdots+X_n^2 & X_1Y_1+X_2Y_2+\cdots+X_nY & X_1+X_2+\cdots+X_n \\ X_1Y_1+X_2Y_2+\cdots+X_nY_n & Y_1^2+Y_2^2+\cdots+Y_n^2 & Y_1+Y_2+\cdots+Y_n \\ X_1+X_2+\cdots+X_n & Y_1+Y_2+\cdots+Y_n & n \end{bmatrix}^{-1} \times \begin{bmatrix} \overline{Y_1}X_1+\overline{Y_2}X_2+\cdots+\overline{Y_n}X_n \\ \overline{Y_1}Y_1+\overline{Y_2}Y_2+\cdots+\overline{Y_n}Y_n \\ \overline{Y_1}+\overline{Y_2}+\cdots+\overline{Y_n} \end{bmatrix}. \quad (10)$$

It seems, at the first glance, that Eq. (10) is a little too complicated for practical implementation. Nevertheless, all the elements in the matrices can be treated as the inner product of two vectors and some of these entries are actually duplicated. This implies that this computation can be efficiently implemented with a fast vector inner product algorithm. Moreover, the involved matrices are only 3×3 and their inverses can be easily computed. Based on a software simulation running on a Pentium III at 450 MHz, the affine coefficients for a single frame can be computed in 5×10^{-6} s. Figure 11 shows the simulation of motion compensation after using the affine model. Figures 11(a) and 11(b) illustrate two consecutive image frames with a rotatory motion. Figure 11(d) shows the detected local motion vectors after motion estimation and motion decision. Based on these valid motion vectors, we calculate the coefficients of the affine model, and Fig. 11(e) shows the stabilized image frame. Figures 11(c) and 11(f) show, respectively, the intensity difference of the two consecutive frames before and after motion compensation.

Moreover, because we must acquire a stabilized image sequence starting from the first frame till the current frame, we accumulate each frame motion vector (FMV) to form an accumulated motion vector (AMV). To suppress error accumulation and to have a mechanism to slowly pull the focus center back to the frame center, the following equation is used to robustly calculate AMV.

$$\mathrm{AMV}[t]=a\times\mathrm{AMV}[t-1]+\mathrm{FMV}[t]. \quad (11)$$

Besides the software simulation of this proposed DIS architecture, a real-time DIS system is also implemented in hardware. Having considered both programming flexibility and hardware efficiency, the motion decision unit and the motion compensation unit are coded in a microprocessor that interconnects with stabilization hardware, which consists of the motion estimation unit and the digital zooming unit. The stabilization hardware is now implemented on an FPGA board. Since the microprocessor is still not powerful enough to support affine modeling, the conventional translational modeling is adopted in this real-time hardware simulation.

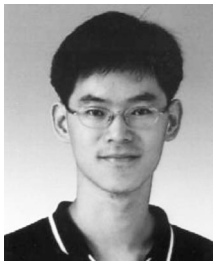
5 Conclusion

In this paper, we propose a fine-division approach over gray-coded bit-planes to achieve high-performance image stabilization. The usage of gray-coded bit-planes greatly reduces the computation complexity of motion estimation, while the fine-division approach improves the usability of

LMVs. In motion decision, a sequence of tests is designed to faithfully extract useful LMVs. These LMVs are then fed into the motion compensation unit to compensate the movement of the whole image. A slightly simplified real-time DIS system is also implemented in hardware, including an FPGA board for the motion estimation unit together with a microprocessor for the motion decision unit and the motion compensation unit.

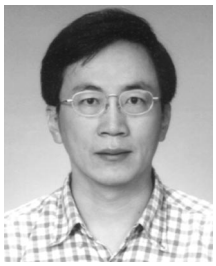
References

1. 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.* **38**(3), 607–616 (1992).
2. T. Kinugasa, N. Yamamoto, and H. Komatsu, ''Electronic image stabilizer for video camera use,'' *IEEE Trans. Consum. Electron.* **36**(3), 520–525 (1990).
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.* **36**(3), 510–519 (1990).
4. 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.* **3**(3), 351–356 (1995).
5. Y. Egusa, H. Akahori, A. Morimura, and N. Wakami, ''An electronic video camera image stabilizer operated on fuzzy theory,'' *Proc. IEEE Cont. Fuzzy Syst*., 851–858 (1992).
6. C. Morimoto and R. Chellappa, ''Evaluation of image stabilization algorithms,'' *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing* **5**, 2789–2792 (1998).
7. M. Sekine, T. Kondou, and H. Hirose, ''Motion vector detecting system for video images stabilizers,'' *IEEE Trans. Consum. Electron.* 268–269 (1994).
8. S.-H. Lee, K.-H. Lee, and S.-J. Ko, ''Digital image stabilizing algorithms based on bit-plane matching,'' *IEEE Int'l. Conf. Consum. Electron*., 126–127 (1998).
9. S.-H. Lee, S.-W. Jeon, E.-S. Kang, and S.-J. Ko, ''Fast digital stabilizer based on gray coded bit-plane matching,'' *IEEE Trans. Consum. Electron.* **45**(3), 598–603 (1999).
10. 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.* **37**(3), 521–530 (1991).



Yeou-Min Yeh received his BS and MS degrees in electronics engineering from National Chiao Tung University, Taiwan, Hsinchu, in 1998 and 2000, respectively. He is currently with ALi (Acer Labs Inc.), Taiwan, Hsinchu. In 2000, he was a part-time development engineer with the Industrial Technology Research Institute. His research interests include digital signal processing, digital image processing, and communications.

Huang-Cheng Chiang received his BS, MS, and PhD degrees from the Department of Electrical Engineering of Tatung University, Taipei, Taiwan, in 1990, 1992, and 1996, respectively. From October 1996 to 1999 he was a research engineer engaged in the development of digital still camera with the Image Technology Department of Opto-Electronics & System Laboratories (OES). He is currently a section manager in OES involved in the research on high definition TV cameras, real-time color imaging, and video signal processing.



Sheng-Jyh Wang received his BS degree in electronics engineering from National Chiao Tung University, Taiwan, in 1984 and his MS and PhD degrees in electrical engineering from Stanford University, in 1990 and 1995, respectively. He is currently an associate professor with the Department of Electronics Engineering, National Chiao Tung University, Taiwan.