Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I
IMECS 2009, March 18 - 20, 2009, Hong Kong

2D One-Bit-Transform Motion Estimation Algorithm with Smoothing and Preprocessing

Wai Chong Chia, Li Wern Chew, Li-Minn Ang, and Kah Phooi Seng

***Abstract*— A high performance 2D one-bit-transform (1BT) motion estimation algorithm with smoothing and preprocessing (S+P) is introduced in this paper. In 1BT motion estimation algorithm, the 8-bit current frame (c frame) and reference frame (p frame) are transformed into their 1-bit representation image, before calculating the Sum of Absolute Difference (SAD) and performing the search operations. For our proposed algorithm, a smoothing threshold ($Threshold_S$) is incorporated into the filtering kernel, which is used to perform the transformation from 8-bit image into its 1BT image. This smoothing technique can greatly reduce the scattering noise created in the 1BT image. After transformation, the 1BT image for the c frame and p frame is divided into macroblocks. The macroblock in the c frame will be first compared to the macroblock at the same position in the p frame. If the SAD is below the preprocessing threshold ($Threshold_P$), the macroblock is considered to have negligible movement and search operation is not required. This preprocessing technique can greatly reduce the total number of search operations. Simulation results show that an improvement up to 0.64 dB, with reduction in search operation up to 93.36% is achieved for video conferencing sequences.**

***Index Terms*— Full search block matching algorithm, motion estimation, one-bit-transform.**

I. Introduction

Motion estimation is a common video compression technique used to exploit the correlated information between the current frame and the reference frame. Among the various types of motion estimation algorithm, the most popular algorithm that provides optimal performance is the Full Search Block Matching Algorithm (FSBMA).

In FSBMA, search operation is performed on every macroblock in the current frame. For each macroblock, searching is conducted within the search window in the reference frame to determine the best matching macroblock. The degree of matching is commonly evaluated by using the Sum of Absolute Difference (SAD), due to its simplicity in implementation. For a block with size of N x N pixels, the SAD can be calculated by (1), whereby c represents the current frame, p represents the reference frame, and i and j represents the coordinate of the image.

Manuscript received September 26, 2008.

W. C. Chia, L. W. Chew, L-M. Ang, and K. P. Seng are with the School of Electrical & Electronic Engineering, The University of Nottingham Malaysia Campus, Jalan Broga, 43500, Semenyih, Selangor, Malaysia (phone: 603-89248000; fax: 603-89248002; correspond e-mail: [keyx7cwc / eyx6clw / kezklma / kezkps] @nottingham.edu.my).

$$SAD(x, y) = \sum_{i=1}^{N-1}\sum_{j=1}^{N-1}\left|c(i, j) - p(i + x, j + y)\right| \tag{1}$$

Although FSBMA provides optimal performance, it is computationally intensive due to the high number of search operations required [1]. According to [5] and [8], 50% to 80% of the video processing time is consumed by the FSBMA. This creates a problem in the era where power consumption has becoming an important design constraint, especially for embedded system [2]-[3].

In order to overcome this problem, many fast motion estimation algorithms such as the New Three Step Search [4], Four Step Search [5], 2D-Logarithmic Search [6] and Conjugate Direction Search [7] have been proposed. These algorithms reduce the number of search points and hence reduce the number of search operations required. However, the performance of these algorithms is only sub-optimal, because there is always a possibility for these algorithms to miss the best matching point.

Other than reducing the number of search operation, simplifying the matching error calculation process is another way to reduce the complexity. The algorithm reported in [8] and [9] uses a technique called one-bit-transform (1BT) to transform an 8-bit image into its 1-bit representation image. Then, the FSBMA and SAD calculations are carried out on the 1BT image. This simplifies the calculation process of SAD down to simple exclusive-OR (XOR) operation as shown in (2), and hence reduces the overall complexity. Other related algorithms are also reported in [10]–[12].

$$SAD(x, y) = \sum_{i=1}^{N-1}\sum_{j=1}^{N-1}\left|c(i, j) \oplus p(i + x, j + y)\right| \tag{2}$$

In this paper, a smoothing and preprocessing (S+P) technique is added to the Multiplication-Free 1BT (MF-1BT) motion estimation algorithm presented in [8]. Although another algorithm based on 1BT is also introduced in [9], it involves floating-point multiplication which is usually slow in hardware and software implementation [9]. Hence, the MF-1BT algorithm is chosen to be the basis of our proposed algorithm. The proposed S+P technique not only maintains the simplicity in the matching error calculations, but also reduces the number of search operations and provides better performance.

Firstly, scattering noise which is a common problem in the 1BT image is reduced by using the proposed smoothing technique. The scattering noise is one of the reasons that lead to performance degradation, since it affects the accuracy in finding the best matching block. Secondly, the large number of search operations in the FSBMA is reduced by using the proposed preprocessing technique. Reducing the number of search operations will shorten the video processing time. Moreover, it can also help to conserve power, as dynamic frequency scaling is now a common technique implemented in many processors.

The paper is organized in the following manner. A brief overview on the MF-1BT motion estimation algorithm is given in section II. Then, the proposed S+P technique is explained in detail in Section III. All the simulation results will be presented and discussed in Section IV. This is follow by the conclusion in Section V.

II. Background

The overall system block diagram of the MF-1BT motion estimation algorithm is shown in Fig. 1. First of all, the current and the reference 8-bit video frames are filtered by using the convolutional kernel (K) proposed in [8]. Then, the current and reference 8-bit video frames are compared with their filtered version respectively. The 1BT image is generated based on the thresholding decision shown in (3), whereby B represents the output 1BT image, I represents the input 8-bit image, I_F represents the filtered version of the input image, and i and j represent the coordinate of the image.

$$B(i,j)=\begin{cases}1, \text{ If } |I(i,j)| \geq |I_F(i,j)| \\ 0, \text{Otherwise}\end{cases} \quad (3)$$

It can be seen that the 1BT image is purely a binary image with the value of ‘0’ and ‘1’ only. Next, the 1BT image of the current and the reference frame is divided into macroblocks respectively, and FSBMA is performed to determine the motion vector for each of the macroblocks. Finally, the motion vector for all the macroblocks are transmitted as output.

III. The Proposed S+P Technique

A. Smoothing Technique

The proposed smoothing technique is incorporated into the convolutional kernel (K) module that is shown in Fig. 1. The purpose of smoothing is to remove the scattering noise that is created in the 1BT image. Fig. 2 shows some sample frames from the video sequence *Claire* that explains the effect of smoothing. From the 8-bit residual image shown in Fig. 2(a), notice that the background is very smooth, and the moving object (*Claire*’s face) that is concentrated at the middle of the frame can be seen clearly.

By comparison, the moving object is difficult to be seen in the 1BT residual image shown in Fig. 2(b), due to the scattering noise. The scattering noise is created by the small difference in pixel values between the original video frame and its filtered version. Although the difference is very small, it can have a significant effect on the SAD calculation in the 1BT image.

For example, consider two pixels A and A’ at the same pixel location in the 8-bit current and reference frame are having the value of 129 and 127 respectively. If a block size of 16 x 16 is used, and all pixels in the block other than the two pixels mentioned above are having the same value, the overall SAD which can be calculated as [(129-127) ÷ (16 x 16 x 255)] x 100%, is only 0.003%.

On the other hand, assuming that the filtered version of pixel A and A’ are having the same value which is 128. The value of pixel A is smaller than its filtered version, whereas the value of pixel A’ is larger than its filtered version. According to the thresholding decision shown in (3), pixel A and pixel A’ will be assigned with ‘0’ and ‘1’ in the 1BT image respectively. Under the same condition applied to the

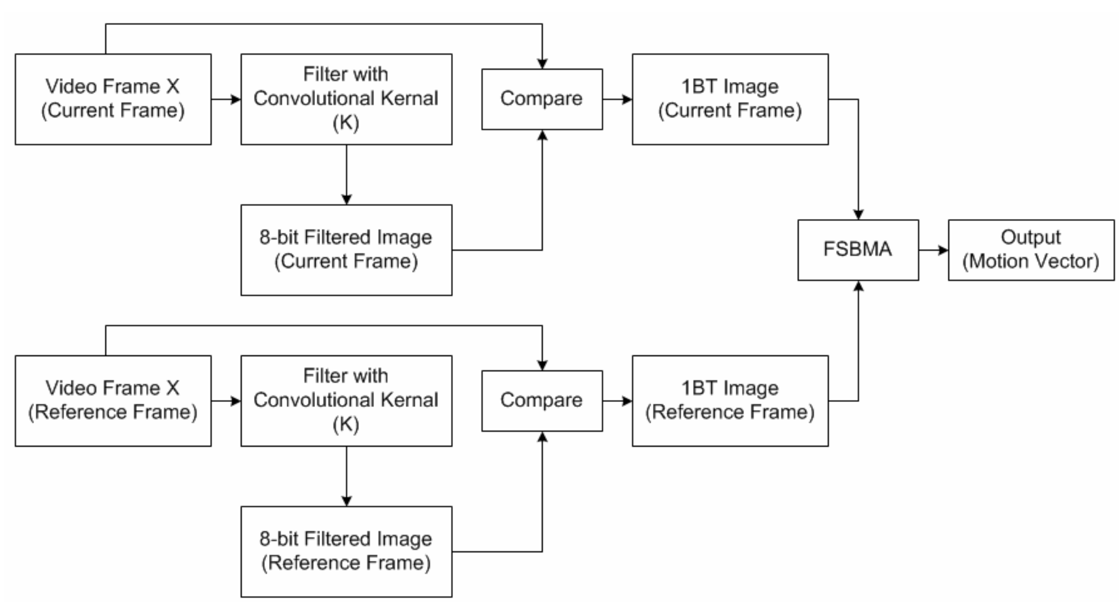

Figure 1. The system block diagram of the MF-1BT motion estimation algorithm.

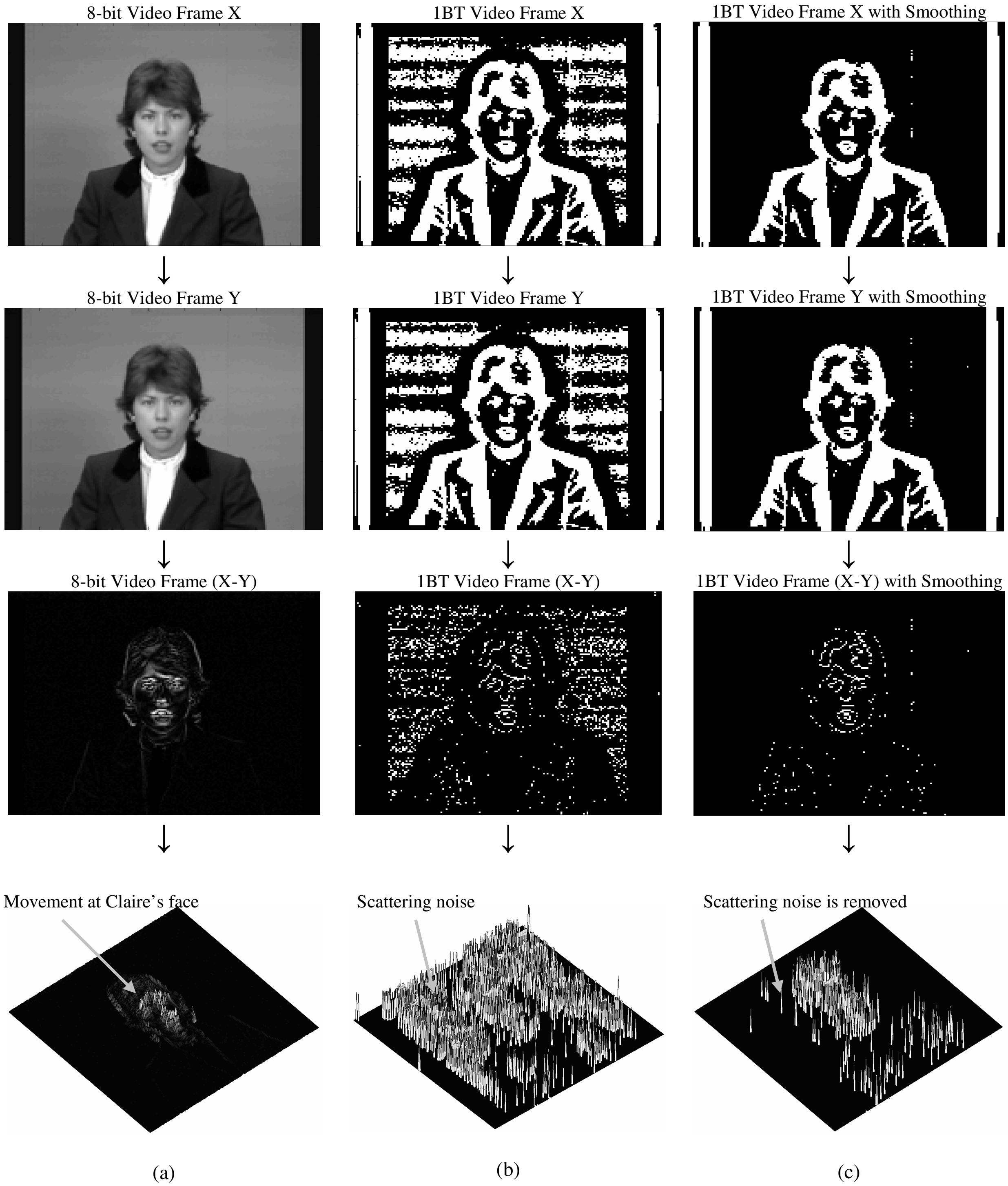

Figure 2. Difference between residual images obtained from (a) 8-bit frame (b) 1BT frame, and (c) 1BT frame with smoothing.

8-bit image, the overall SAD which can be calculated as [(1-0) ÷ (16 x 16 x 1)] x 100%, is increase to 0.39%. It should be noted that the maximum difference of pixel value in 1BT image is 1, instead of 255 in 8-bit image. This shows that a small difference in the pixel value can affect the SAD significantly.

In order to overcome this problem, it is necessary to prevent the small changes in pixel values from affecting the decision output of the 1BT image. This is the main reason that leads to the introduction of the smoothing threshold (Threshold_S) into the convolutional kernel (K), which is used to transform an 8-bit image into its 1BT image. By adding the smoothing threshold into (3), the thresholding decision is altered and shown in (4).

$$B(i,j)=\begin{cases}1, \text{ If } |I(i,j)| \geq |I_F(i,j)| + \text{Threshold}_S \\ 0, \text{ Otherwise}\end{cases} \tag{4}$$

The effect of applying the smoothing technique can be seen clearly from the 1BT residual image shown in Fig. 2(c). Notice that the scattering noise in the background is greatly reduced. Furthermore, it is now easier to locate the moving object within the video frame. By reducing the scattering noise, the SAD calculation becomes more accurate and the efficiency in finding the best matching macroblock is improved.

B. Preprocessing Technique

The preprocessing technique is added to the FSBMA module that is shown in Fig. 1 to reduce the number of search operations. In many situations, the background or certain object in a video sequence is usually static or has negligible movement. For example, it can be seen that the movement in video sequence *Claire* is mostly concentrated in the middle of the frame. Therefore, it is a waste to perform FSBMA for the entire video frame, due to the high probability that the best matching block is located at the same position in the reference frame.

The idea of preprocessing is to prevent the FSBMA being performed on macroblocks which have negligible movement. This helps to shorten the video processing time and reduce power consumption. Algorithm I describes the preprocessing steps that will be carried out before FSBMA is performed on a macroblock.

For each of the macroblocks in the current frame, it will be first compared with the macroblock at the same position in the reference frame. Then, the SAD is computed using (2) and compared to the preprocessing threshold ($Threshold_P$). If the SAD is smaller than or equal to $Threshold_P$, it is presumed that the macroblock has negligible movement and no search operation is required. Otherwise, FSBMA is performed to find the best matching block in the reference frame. This process will be repeated for all macroblocks.

<u>Algorithm I</u>
For each of the macroblocks,
1. Compute SAD (0, 0) using (2).
2. If SAD (0, 0) $\leq$ $Threshold_P$, halt.
3. If SAD (0, 0) $>$ $Threshold_P$, perform FSBMA.

IV. Simulation Results and Discussions

Simulation of the proposed S+P technique is performed on six video sequences, which include *Claire*, *Miss America*, *Akiyo*, *Foreman*, *Football*, and *Mobile*. Only the luminance (Y) component is considered in the simulation, and a block size of 16 x 16 pixels with a search window of 16 pixels is used. The motion vectors generated are used to reconstruct the video frame, and its quality is evaluated by calculating the peak signal-to-noise-ratio (PSNR) in decibel (dB).

A. Determination of $Threshold_S$ and $Threshold_P$

Before the simulation is performed, it is necessary to determine the value of $Threshold_S$ and $Threshold_P$. Hence, an analysis is carried out to determine the suitable value of $Threshold_S$ and $Threshold_P$. In the analysis, the proposed algorithm is applied to the first 20 frames of all the video sequences. Then, the values of $Threshold_S$ and $Threshold_P$ are varied from 0 to 10 and 2 to 12 respectively. Fig. 3 shows the plot of PSNR versus $Threshold_S$ for different value of $Threshold_P$.

From the plot, it was found that the best choice for $Threshold_S$ and $Threshold_P$ are 3 and 10 respectively. In fact, the optimum threshold for each video sequence is about ±1 to ±2 from the value we have chosen. If the value of 2 instead of 3 is used for $Threshold_S$, the degradation in performance for *Claire* is much larger than the gain we can obtained for *Akiyo* and *Foreman*. On the other hand, if the value larger than 3 is used for $Threshold_S$, the performance starts to drop for *Mobile*, *Football*, and *Foreman*. Hence, the value of 3 is chosen for $Threshold_S$. Under the similar observation as $Threshold_S$, the value of 10 is chosen for $Threshold_P$. The main criteria is to select a value of $Threshold_S$ and $Threshold_P$ that can work for most of the situation.

However, it is important to note that there is a possibility that a better performance can be achieved for different video sequences with other threshold values. By assuming that the remaining frames are following the same trend as the first 20 frames, these two threshold values are applied to the entire sequence.

B. Simulation Results

The average PSNR for each video sequence is recorded in Table 1. The proposed algorithm performs very well in lower motion video such as *Claire*, *Miss America*, and *Akiyo*. Most significantly, the average PSNR achieved in these three video sequences is very close to the optimal performance of 8-bit FSBMA. The average PSNR is just 0.1 to 0.41 dB lower than 8-bit FSBMA. But for the MF-1BT, the average PSNR is 0.08 to 1.05 dB lower than 8-bit FSBMA.

On the other hand, the improvement for moderate motion video such as *Foreman*, and high motion video such as *Football* and *Mobile* is not as significant as lower motion video. For *Foreman*, an average gain of 0.22 dB is achieved. A sample reconstructed frame shown in Fig. 4 shows that the visual quality is improved by the proposed S+P technique. It can be seen that the "helmet" is better reconstructed. But for *Football* and *Mobile*, the proposed algorithm is 0.07 dB and 0.03 dB lower than the MF-1BT algorithm. However, the degradation is almost negligible when compared to the gain achieved in other video sequences. Other than performance, the reduction in number of search operations achieved by the proposed algorithm should also be taken into account.

C. Reduction in Number of Search Operations

The reduction in the number of search operations for each video frame is calculated by using (4), whereby N_{FSBMA} represents the total number of search operations required in FSBMA, and N_{S+P} represents the number of search operations carried out when the proposed S+P technique is used. First, the reduction in the number of search operations for all the frames is computed. Then, it is averaged up to obtain the final search reduction for the entire video sequence that is recorded in Table 2.

$$\text{Reduction (\%)} = \frac{N_{FSBMA} - N_{S+P}}{N_{FSBMA}} \times 100\% \quad (4)$$

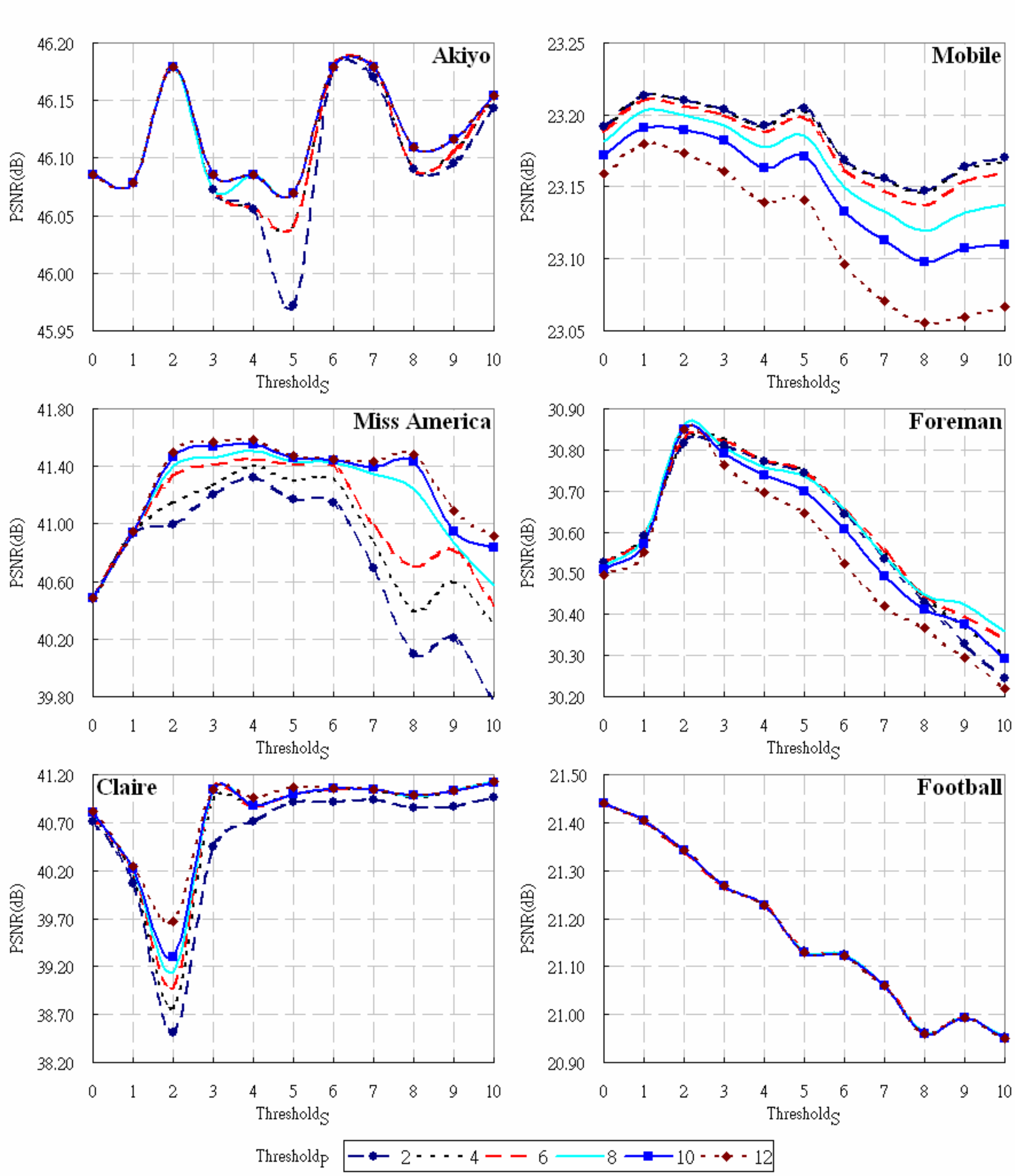

Figure 3. The average PSNR (dB) versus Threshold$_S$ for different value of Threshold$_P$ in the first 20 frames of various video sequences.

Since the algorithm proposed in [8] and [9] applied FSBMA on all the macroblocks in a video frame, the search reduction is zero when compared to 8-bit FSBMA. For lower motion video sequences such as *Claire*, *Miss America*, and *Akiyo*, the number of search operations are significantly reduced by 92.51%, 58.06% and 93.36% respectively when the proposed S+P technique is incorporated.

On the other hand, the reduction in number of search operations for *Foreman*, *Football*, and *Mobile* are 12.13%, 2.03% and 5.11% respectively. Although the proposed S+P technique causes a minor degradation in performance for *Football* and *Mobile*, but in return the number of search operations is reduced for these two video sequences. By reducing the number of search operations required, the video overall processing time and power consumption can also be reduced.

V. Conclusions

Implementation of the S+P technique into the MF-1BT motion estimation algorithm produces good results for lower motion video such as *Claire*, *Miss America* and *Akiyo*. The performance is approaching the optimal 8-bit FSBMA. Furthermore, the number of search operations is greatly reduced, and it also helps to reduce the video processing time and power consumption. Simulation results show that the proposed S+P method can improve the performance up to 0.64 dB, and reduce the number of search operations up to 93.36%. Overall, the proposed S+P method provides an efficient solution for the implementation of motion estimation algorithms under hardware constraint environments. This S+P technique is suitable for video conferencing application which involves lower motion and static background.

Table 1. Average PSNR (dB) of several video sequences reconstructed by using different motion estimation algorithm.

Method	Video Sequence (Frame Size, Sequence Length) $Threshold_S = 3$ & $Threshold_P = 10$					
	Claire (176 x 144) (493 Frames)	Miss America (176 x 144) (149 Frames)	Akiyo (176 x 144) (299 Frames)	Foreman (352 x 288) (399 Frames)	Football (352 x 240) (125 Frames)	Mobile (352 x 288) (299 Frames)
8-bit FSBMA	42.90	41.34	44.30	31.55	22.89	24.59
1BT [6]	42.08	40.44	44.20	29.85	21.80	24.20
MF-1BT [5]	42.09	40.29	44.22	29.92	21.79	24.24
Proposed Algorithm	42.71	40.93	44.20	30.14	21.70	24.21

Table 2. Average search reduction (%) from FSBMA for different motion estimation algorithm in several video sequences.

Method	Video Sequence (Frame Size, Sequence Length) $Threshold_S = 3$ & $Threshold_P = 10$					
	Claire (176 x 144) (493 Frames)	Miss America (176 x 144) (149 Frames)	Akiyo (176 x 144) (299 Frames)	Foreman (352 x 288) (399 Frames)	Football (352 x 240) (125 Frames)	Mobile (352 x 288) (299 Frames)
8-bit FSBMA	-	-	-	-	-	-
1BT [6]	0	0	0	0	0	0
MF-1BT [5]	0	0	0	0	0	0
Proposed Algorithm	92.51	58.06	93.36	12.13	2.03	5.11

(a) (b)

Figure 4. A sample reconstructed video frame from *Foreman* (a) with and (b) without using the proposed S+P technique.

REFERENCES

[1] V. L. Do and K. Y. Yun, "A low-power VLSI architecture for full-search block-matching motion estimation algorithm," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 4, August 1998, pp. 393 – 398.

[2] C. Piguest, *Low-Power Electronics Design*. CRC Press, 2005, p. 36-15.

[3] R. Li, B. Zeng and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuit Syst. Video Technology*., vol. 4, no. 4, Aug. 1994, pp. 438-442.

[4] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuit Syst. Video Technol*ogy, vol. 6, no. 3, June 1996, pp. 313-317.

[5] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun*., vol. COM-29, no. 12, Dec. 1981, pp. 1799-1808.

[6] R. Srinivasan and K. R. Rao, "Predictive Coding Based on Efficient Motion Estimation," *IEEE Trans. Commun*., vol. COM-33, no. 8, Aug. 1985, pp. 888-896.

[7] S. Erturk, "Multiplication-free one-bit transform for low-complexity bock-based motion estimation," *IEEE Signal Process. Lett*., vol. 14, no. 2, Feb. 2007, pp. 109-112.

[8] B. Natarajan, V. Bhaskaran and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *IEEE Trans. Circuit Syst. Video Technology*, vol. 7, no. 4, Aug. 1997, pp. 702-706.

[9] A. Erturk and S. Erturk, "Two-bit transform for binary block motion estimation," *IEEE Trans. Circuits Syst. Video Technology*, vol. 15, no. 7, Jul. 2005, pp. 938-946.

[10] J. Feng, K. T. Lo, H. Mehrpour and A. E. Karbowiak, "Adaptive block matching motion estimation algorithm using bit-plane matching," in *IEEE Int. Conf. Image Processing*, Washington, DC, 1995, pp. 496-499.

[11] M. M. Mizukim, U. Y. Desai, I. Masaki and A. Chandrakasan, "A binary block matching architecture with reduced power consumption and silicon area requirements," in *IEEE ICASSP-96*, vol. 6, Atlanta, 1996, pp. 3248-3251.

[12] X. Xia and Y. Q. Shi, "A thresholding hierarchical block matching algorithm," in *Proc. IEEE Int. Symp. Circuits Syst*., vol. 2, May. 1996, pp. 624-627.