Fast Binary Motion Estimation for Screen Content Video Coding

Ting Sun, Pengfei Wan, Oscar C. Au, Wei Dai, Luheng Jia, Yuan Yuan, Amin Zheng, Rui Ma
The Hong Kong University of Science and Technology, Hong Kong.
E-mail: {tsun, leoman, eeau, weidai, ljia, yyuanad, amzheng, rmaaa}@ust.hk

***Abstract*—One-bit transform (1BT), followed by binary motion estimation, is an effective alternative for accelerating traditional 8-bit motion estimation (ME) in video coding. The underlining assumption in the design of 1BT methods is that natural videos contain noise. For screen content videos, however, the special characteristics (*e.g.* screen content is typically noise-free) can be exploited to further improve the motion estimation accuracy. In this paper we propose a binary ME method which is specifically designed for screen content videos. In particular, binary ME is performed on a selected bit-plane. Our contribution is two-fold: 1) our bit-plane selection is hardware-friendly and content-adaptive to image blocks; 2) a zero-bias early termination scheme is proposed to accelerates the binary ME procedure. Experimental results demonstrate that our proposed binary ME method improves the accuracy of obtained motion vectors for screen content video sequences.**

I. Introduction

Video compression is a necessary step for efficient transmission and storage of digital video contents. In general, compression can be achieved by reducing the redundant information in temporal and spatial domains. Motion estimation (ME) is the main technique to exploit temporal correlation between frames. In ME, block matching (BM) algorithm is commonly used to choose "best" prediction from previously encoded reference frames. In BM, the current frame to be encoded is typically divided into non-overlapping rectangular or square blocks, the predictor of each block is obtained by block-matching within a search range in the reference frame based on some distortion metric, *e.g.* sum of absolute differences (SAD) or sum of squared differences (SSD), etc. The best predictor is the block in the reference frame achieving minimum distortion. The predictor is indicated by a motion vector (MV) (mv_x, mv_y) which denotes the horizontal and vertical location difference from the predictor to the block to be encoded. Thus, ME can be formulated as follows:

$$\begin{aligned} \min_{(mv_x, mv_y)} \quad & \sum_{i=1}^{M}\sum_{j=1}^{N} |\mathbf{I}^t(i,j) - \mathbf{I}^{t-1}(i+mv_x, j+mv_y)| \\ \text{s. t.} \quad & -s \leq mv_x, mv_y \leq s \end{aligned} \tag{1}$$

where in this example SAD is used as distortion metric, t is time index, $\mathbf{I}^t$ is the current frame to be encoded, $\mathbf{I}^{t-1}$ is the reference frame, (i,j) denotes a pixel location, the block size is $M \times N$ and the search range is $-s \leq mv_x, mv_y \leq s$.

The most reliable way to obtain the MV of truly best predictor is to conduct full search (FS), calculating SAD at every location within the search range to find the block given minimum distortion. Since FS is very computation-intensive, a lot of fast ME algorithms have been proposed, including integer-pixel ME as well as sub-pixel ME. For integer-pixel ME, those approaches can be categorized into three main classes: 1) designing some "smart" search patterns, often equipped with early termination, such as three step search [1], new three step search [2], diamond search [3], 2-D logarithmic search [4], UMHexagon search [5], EPZS [6], PMVFAST [7], E-PMVFAST [8] etc [9–12], but complex search patterns are not hardware friendly; 2) using FS-ME while adaptively changing the search window size according to the motion properties of the current block [13, 14]; 3) simplifying the cost function computation, such as successive elimination algorithm (SEA) [15, 16], partial distortion search (PDS) [17], one-bit transform [18–21] and two-bit transform [22].

FS costs a lot of computation power for frames with 8-bit-depth pixel precision. For binary frames, however, FS can be done much faster since the SAD between two binary blocks is simply Hamming distance which can be calculated using *exclusive-or* (XOR) instead of integer subtraction. Motivated by the low-complexity and hardware-friendly implementation of exclusive-or, various one-bit transform (1BT) techniques were proposed [18–21, 23]. The fact is that traditional 1BT methods are mostly designed for compression of natural video sequences. With the increasing need of compression of *screen content videos*, the traditional 1BT ME methods can be sub-optimal because screen content possesses distinct characteristics from natural videos. Thus in this paper we tailor the binarization method for screen content videos to improve ME accuracy.

The rest of the paper is organized as follows. The 1BT method is reviewed in Section II, followed by our proposed binary ME method in Section III. Experimental results are shown in Section IV. Section V summarizes our work.

II. One-bit transform

1BT is an image binarizaton method which is proposed as a preprocessing to enable binary ME in video coding [18, 19]. The 8-bit frame $\mathbf{I}$ is first convolved with kernel $\mathbf{K}$ in (2) and the resultant frame $\hat{\mathbf{I}}$ is used as the per-pixel threshold to binarize the collocated pixel in $\mathbf{I}$ following (3). The whole procedure of 1BT in [19] is shown in Fig. 1.

$$\mathbf{K}(i,j) = \begin{cases} \frac{1}{25} & \text{if } i,j \in [0,4,8,12,16] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$
\mathbf{B}(i,j)=\begin{cases}1 & \text{if } \mathbf{I}(i,j)\geq \hat{\mathbf{I}}\\ 0 & \text{otherwise}\end{cases} \tag{3}
$$

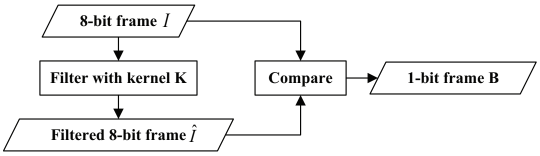


Fig. 1. Block Diagram of 1BT

After the reference frame and current frame to be encoded are converted into binary frames $\mathbf{B}^{t-1}$ and $\mathbf{B}^{t}$ respectively, the binary ME is conducted to determine the predictor:

$$
\begin{aligned}
\min_{(mv_x,mv_y)} \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}\mathbf{B}^{t}(i,j)\oplus\mathbf{B}^{t-1}(i+mv_x,j+mv_y)\\
\text{s. t.} \quad & -s\leq mv_x,mv_y\leq s
\end{aligned} \tag{4}
$$

where $\oplus$ denote Boolean XOR operator and the superscript t and $t-1$ are time index for current frame and reference frame.

The assumption of 1BT is that the high frequency edge information is the key hint for ME, but the noise in natural videos also exhibit itself as high frequency components, so the band-pass filter in (2) is applied [18] to extract useful edge information. Therefore traditional 1BT method is suitable for natural videos. For screen content videos which are typically noise-free, 1BT can still be applied with reasonable performance, but we claim that a better binarization scheme is possible if we explicitly exploit the special characteristics of screen content video. Another problem of traditional 1BT method is that calculation of binarization threshold still involves 8-bit numbers and is sensitive to a few extreme value pixels. Similar pixels may be binarized to different results just because few extreme value pixels participate in one of the threshold calculation.

Next we present our proposed binary ME method which is specifically tailored for screen content video.

III. The Proposed Fast Binary ME for Screen Content Video

Proposed fast binary ME is composed of mainly two steps: adaptive bit-plane selection and binary ME on the selected bit-plane. Instead of using reconstructed frames, we propose to use original noise-free frames as reference. The reasons for using original frame as reference frame are: 1) after binarization, there's no guarantee that the MVs obtained by binary ME on reconstructed frame will lead to least residual energy; 2) original 8-bit frame is not contaminated by quantization noise, so its binarized frame faithfully preserves the useful structures (*e.g.* edges) of the screen image content.

A. Adaptive Bit-plane Selection

Our goal is to convert the 8-bit original frames into binary ones, such that the MVs obtained by binary ME are close to the ground-truth (those obtained by traditional FS).

1) Algorithm description: A 8-bit frame $\mathbf{I}$ can be decomposed into eight bit-planes denoted by $\mathbf{B}_0,\mathbf{B}_1,\ldots,\mathbf{B}_7$, where $\mathbf{B}_7$ is the most significant bit-plane and $\mathbf{B}_0$ is the least significant one. The absolute difference between two 8-bit pixels located at (i,j) and (i',j') can be written as:

$$
\begin{aligned}
&|\mathbf{I}^{t}(i,j)-\mathbf{I}^{t-1}(i',j')|\\
=&|\sum_{k=0}^{7}\mathbf{B}_k^{t}(i,j)\times 2^k-\sum_{k=0}^{7}\mathbf{B}_k^{t-1}(i',j')\times 2^k|\\
=&|\sum_{k=0}^{7}(\mathbf{B}_k^{t}(i,j)-\mathbf{B}_k^{t-1}(i',j'))\times 2^k|
\end{aligned} \tag{5}
$$

It is easy to verify that the difference in more significant bit (MSB) contributes more to the distortion calculation. The basic idea of our adaptive selection is that for each block, only one bit-plane is used for binary ME and *higher priority is given to more significant bit-planes*. Lower bit planes are considered only when higher bit planes fail to offer enough image structures for ME. The block diagram of our proposed binary ME is shown in Fig 2.

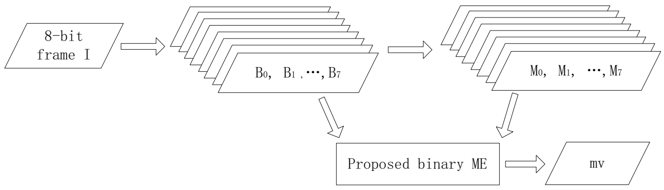


Fig. 2. Block diagram of proposed binary ME

One binary edge-map $\mathbf{M}_k$ is generated from one bit-plane $\mathbf{B}_k$, in which the pixels next to an edge is 1 and 0 otherwise.

$$
\mathbf{M}_k(i,j)=\begin{cases}1 & \text{if } \exists(m,n)\in\{(\pm1,0),(0,\pm1)\}\text{, such that:}\\ & \mathbf{B}_k(i,j)\oplus\mathbf{B}_k(i+m,j+n)=1\\ 0 & \text{otherwise}\end{cases} \tag{6}
$$

where (i,j) denotes pixel position and $k\in\{0,1,\ldots,7\}$ is the bit plane index. Based on the 8 binary edge-maps, next we select one bit-plane on which binary ME will be performed for the current block. In particular, Algorithm 1 summaries our adaptive bit-plane selection, where $\mathbf{C}_k^t$ counts the number of ones in a block of $\mathbf{M}_k^t$ located at (i,j), the block size is $N\times N$, T denotes a threshold which is set to $2N$ and selected_bp$(i,j)\in\{0,1,\ldots,7\}$ is used to store the selection of bit-plane for current block at (i,j).

2) Hardware-friendly Implementation: The edge-map generation in (6) only involves binary operations, thus we present a hardware-friendly implementation in Algorithm 2. Specifically, we use $\mathcal{S}_{\text{left}}(\cdot)$, $\mathcal{S}_{\text{right}}(\cdot)$, $\mathcal{S}_{\text{top}}(\cdot)$, $\mathcal{S}_{\text{down}}(\cdot)$ to denote the left, right, top and bottom spacial shift operations respectively. Except for the pixel (i,j) at image boundary, we have: $\mathcal{S}_{\text{left}}(I)(i,j)=I(i,j+1),\mathcal{S}_{\text{right}}(I)(i,j)=I(i,j-1),\mathcal{S}_{\text{top}}(I)(i,j)=I(i+1,j),\mathcal{S}_{\text{down}}(I)(i,j)=I(i-1,j)$.

In practice, the register may not be large enough to read in the whole frame at one time, but Algorithm 2 can be easily

Algorithm 1 Adaptive Bit-plane Selection

Input: $\mathbf{M}_0^t, \ldots \mathbf{M}_7^t$
Output: selected_bp

selected_bp$(i,j) \leftarrow 0$ for all (i,j) ▷ initialization
calculate $\mathbf{C}_k^t(i,j)$ for all k and (i,j)
for all blocks located at (i,j) **do**
  for integer k from 7 to 0 **do**
    if $\mathbf{C}_k^t(i,j) \geq T$ **then**
      selected_bp$(i,j) \leftarrow k$ ▷ select bit-plane
      break
    end if
  end for
end for

Algorithm 2 Frame-based Mask Generation

Input: B_k
Output: M_k

$M_k^{\text{left}} = B_k \oplus \mathcal{S}_{\text{left}}(B_k)$
$M_k^{\text{right}} = \mathcal{S}_{\text{right}}(M_k^{\text{left}})$
$M_k^{\text{top}} = B_k \oplus \mathcal{S}_{\text{top}}(B_k)$
$M_k^{\text{down}} = \mathcal{S}_{\text{down}}(M_k^{\text{top}})$
$M_k = M_k^{\text{left}} | M_k^{\text{right}} | M_k^{\text{top}} | M_k^{\text{down}}$ ▷ entrywise OR

extended to block-based implementation by simply replace B_k by a block. Regarding the computational complexity, we see 1) in 1BT, band-pass filter requires $M \times N \times 25$ integer addition and 1 division, while binary block generation (3) requires $M \times N$ integer subtraction (comparison); 2) in proposed method, the edge-map generation requires $M \times N \times 16$ XOR and $M \times N \times 3 \times 8$ OR operations, while bit-plane selection requires $M \times N$ to $(M \times N) \times 7$ integer additions. Since Boolean operation can be done much faster than integer operation and (6) requires only 3×3 window while 2 in 1BT requires 17×17 window so the proposed method is much more hardware-friendly.

B. Binary ME with Zero-bias Early Termination

Due to the nature of screen content, there exist large amount of zero-motion blocks in screen content videos. A typical example is shown in Fig. 3, which plots the histogram of $|mv_x| + |mv_y|$ obtained by traditional FS on the first 100 frames of screen content video `SlideShow`. Besides the dominant percentage of zero-motion blocks, the zero-motion blocks usually have no change in pixel intensities as well (*e.g.* the background of powerpoint slides). Based on these observations, we propose a simple yet effective zero-bias early termination to further accelerate traditional binary ME using full search. In particular, early termination is triggered when the prediction residual R satisfies $||R|| = 0$ at $(mv_x, mv_y) = (0,0)$. This is different from other non-zero threshold early termination, since $||R|| = 0$ ensures the best predictor if triggered, *i.e.* proposed early termination accelerates ME without degrading the reliability of the resultant MVs at all. It's worth mentioning that proposed early termination may not occur often in natural video sequences due to noise and camera motion. However, for noise-free screen content videos, the exact matching can be found with very high chance (to be discussed in Section IV).

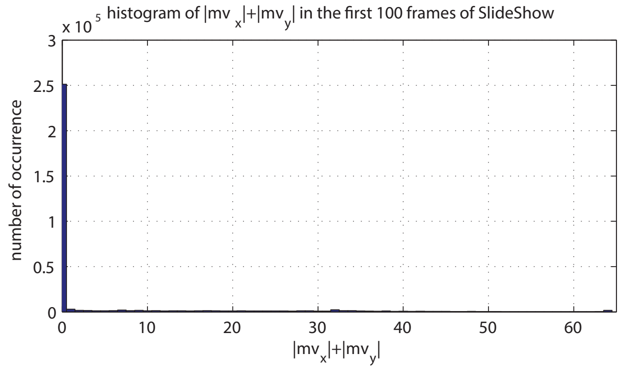
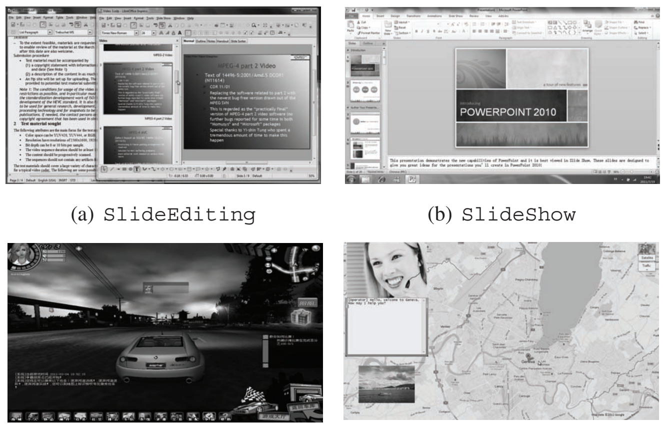


Fig. 3. Histogram of $|mv_x| + |mv_y|$ of all the best predictors obtained by FS from the first 100 frames of video sequence `SlideShow`

IV. Experimental results

We conduct our experiments using matlab and we test the performance of the following binary ME methods: **1BT** [19] is the traditional 1BT followed by binary ME; **1BTO** is the same as **1BT** except that it use original frames as reference instead of the reconstructed ones; **Prop** is the proposed binary ME method. We use **FS** to denote traditional 8-bit full search using reconstructed frames.

Test sequences are 4 typical screen content video sequences `SlideEditing`, `SlideShow`, `ChinaSpeed` and `map`, see Fig. 4. Without loss of generality, we only test the first 100 Y-frames. The block size for ME is fixed to be 16×16 and the search range is set to $-32 \leq mv_x, mv_y \leq 32$.



(a) `SlideEditing` (b) `SlideShow`

(c) `ChinaSpeed` (d) `map`

Fig. 4. Four test sequences used in our experiments

We measure the performance of each binary ME method using the following 3 metrics: average SAD of the obtained MVs (SAD-MV), average SAD of the prediction residuals (SAD-RES) and correct MVs ratio. For SAD-MV, we use the MVs $(mv_x^{\mathrm{FS}}, mv_y^{\mathrm{FS}})$ obtained by FS in 8-bit-depth frames as ground truth. Specifically, SAD-MV is calculated as a summation over all blocks within a frame:

$$\sum_{i,j} |mv_x(i,j) - mv_x^{\mathrm{FS}}(i,j)| + |mv_y(i,j) - mv_y^{\mathrm{FS}}(i,j)| \quad (7)$$

Table I, Table II and Table III show the SAD-MV, SAD-RES and ratio of correct MVs (averaged over 100 frames) respectively, with the best results highlighted in bold font. We see that proposed method achieves the best performance in almost all experiments.

TABLE I
AVERAGE SAD-MV IN ONE FRAME

	SlideEditing	SlideShow	ChinaSpeed	map
1BT	30125	14502	43781	10638
1BTO	19738	12039	39116	4432.1
Prop	**19290**	**10778**	**34201**	**4297.2**

TABLE II
AVERAGE SAD OF RESIDUE IN ONE FRAME

	SlideEditing	SlideShow	ChinaSpeed	map
FS	1279300	5447000	2021200	1128200
1BT	2239500	7934000	4214200	1452800
1BTO	1610100	7550700	3746700	1255700
Prop	**1557200**	**6209100**	**3538500**	**1238400**

TABLE III
RATIO OF CORRECT MVS

	SlideEditing	SlideShow	ChinaSpeed	map
1BT	0.7473	0.8336	0.4296	0.8856
1BTO	0.8357	0.8657	0.4817	**0.9541**
Prop	**0.8359**	**0.8697**	**0.4922**	0.9529

To justify the effectiveness of proposed zero-bias early termination, we test the percentage of blocks that trigger our zero-bias early termination. We see from Table IV that in `SlideEditing`, `SlideShow` and `map`, more than 70% of binary full search are avoided by **1BTO** and **Prop**. This percentage decreases dramatically for `ChinaSpeed`, since this sequence mimic the real scene of driving so there is a lot of motion.

TABLE IV
PERCENTAGE OF BLOCKS TRIGGER ZERO-BIAS EARLY TERMINATION IN A FRAME

	SlideEditing	SlideShow	ChinaSpeed	map
FS	8.17%	56.44%	0.46%	16.51%
1BT	23.11%	72.31%	4.92%	16.09%
1BTO	88.11%	96.23%	10.35%	94.87%
prop	76.25%	94.44%	6.69%	94.86%

Comparing the performance between **1BT** and **1BTO**, we see that the accuracy of MVs improves and the energy in residue decreases a lot by using the original frames as reference instead of the reconstructed ones. Further more, the quantization error greatly degrades the effectiveness of zero-bias early termination method, which means binary ME using reconstructed reference frames consumes much more time. Comparing the performance of **1BTO** and **Prop**, we see that proposed pyramid binary ME strategy obtain the MVs closer to the MVs obtained by traditional FS with smaller energy in residue as well.

V. CONCLUSION

We propose a new binary ME method for screen content videos. Our method contains two steps: adaptive bit-plane selection and binary full search with zero-bias early termination. Experimental results show that the proposed method increases the accuracy of resultant MVs and zero-bias early termination effectively avoids unnecessary full search in binary ME.

VI. ACKNOWLEDGEMENT

This work is supported in part by Hong Kong Govt Innovation and Technology Fund and State Key Laboratory on Advanced Displays and Optoelectronics Technologies (Project No: ITC-PSKL12EG02), and by HKUST (HKUST Project no. FSGRF12EG01 and FSGRF14EG40).

REFERENCES

[1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion-compensated interframe coding for video conferencing,” in *Proc. NTC 81*, 1981, p. 9.

[2] R. Li, B. Zeng, and M. L. Liou, “A new three-step search algorithm for block motion estimation,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 4, no. 4, pp. 438–442, 1994.

[3] Shan Zhu and Kai-Kuang Ma, “A new diamond search algorithm for fast block-matching motion estimation,” *Image Processing, IEEE Transactions on*, vol. 9, no. 2, pp. 287–290, 2000.

[4] J. Jain and A. Jain, “Displacement measurement and its application in interframe image coding,” *Communications, IEEE Transactions on*, vol. 29, no. 12, pp. 1799–1808, 1981.

[5] Xie Lifen, Huang Chunqing, and Chen Bihui, “Umhexagons search algorithm for fast motion estimation,” in *Computer Research and Development (ICCRD), 2011 3rd International Conference on*. IEEE, 2011, vol. 1, pp. 483–487.

[6] Alexis M Tourapis, “Enhanced predictive zonal search for single and multiple frame motion estimation,” in *Electronic Imaging 2002*. International Society for Optics and Photonics, 2002, pp. 1069–1079.

[7] A. M. Tourapis, O. C. Au, and M. L. Liou, “Predictive motion vector field adaptive search technique (PMVFAST)-enhancing block based motion estimation,” in *Proceedings of SPIE*, 2001, vol. 4310, pp. 883–892.

[8] H. M. Wong, O. C. Au, C. W. Ho, and S. K. Yip, “Enhanced predictive motion vector field adaptive search technique (E-PMVFAST)-based on future mv prediction,” in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 4–pp.

[9] Wei Dai, O.C. Au, Sijin Li, Lin Sun, and Ruobing Zou, “Fast sub-pixel motion estimation with simplified modeling in hevc,” in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, May 2012, pp. 1560–1563.

[10] Wei Dai, O.C. Au, Chao Pang, Lin Sun, Ruobing Zou, and Sijin Li, “A novel fast two step sub-pixel motion estimation algorithm in hevc,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 1197–1200.

[11] J.W. Suh, J Cho, and J Jeong, “Model-based quarter-pixel motion estimation with low computational complexity,” *Electronics letters*, vol. 45, no. 12, pp. 618–620, 2009.

[12] J. F. Chang and J. J. Leou, “A quadratic prediction based fractional-pixel motion estimation algorithm for H.264,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 5, pp. 1074–1089, 2006.

[13] Z. Chen, Q. Liu, T. Ikenaga, and S. Goto, "A motion vector difference based self-incremental adaptive search range algorithm for variable block size motion estimation," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 1988–1991.

[14] W. Dai, O. C Au, S. Li, L. Sun, and R. Zou, "Adaptive search range algorithm based on cauchy distribution," in *Visual Communications and Image Processing (VCIP), 2012 IEEE*. IEEE, 2012, pp. 1–5.

[15] Wenhua Li and Ezzatollah Salari, "Successive elimination algorithm for motion estimation," *Image Processing, IEEE Transactions on*, vol. 4, no. 1, pp. 105–107, 1995.

[16] Tae Gyoung Ahn, Yong Ho Moon, and Jae Ho Kim, "Fast full-search motion estimation based on multilevel successive elimination algorithm," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 11, pp. 1265–1269, 2004.

[17] Xuan Jing and Lap-Pui Chau, "Partial distortion search algorithm using predictive search area for fast full-search motion estimation," *Signal Processing Letters, IEEE*, vol. 14, no. 11, pp. 840–843, 2007.

[18] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 7, no. 4, pp. 702–706, Aug 1997.

[19] P.H.-W. Wong and O.C. Au, "Modified one-bit transform for motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, no. 7, pp. 1020–1024, Oct 1999.

[20] Xudong Song, Tihao Chiang, X. Lee, and Ya-Qin Zhang, "New fast binary pyramid motion estimation for mpeg2 and hdtv encoding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, no. 7, pp. 1015–1028, Oct 2000.

[21] O. Urhan and S. Erturk, "Constrained one-bit transform for low complexity block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 4, pp. 478–482, April 2007.

[22] A. Erturk and S. Erturk, "Two-bit transform for binary block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 7, pp. 938–946, July 2005.

[23] Jeng-Hung Luo, Chung-Neng Wang, and Tihao Chiang, "A novel all-binary motion estimation (abme) with optimized hardware architectures," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 8, pp. 700–712, Aug 2002.