

Binary Adaptive Luminance Mapping for Motion Estimation

Xianghu Ji, Jie Liu, Chuang Zhu, Huizhu Jia, Xiaodong Xie, and Wen Gao

National Engineering Laboratory for Video Technology, Peking University
Beijing 100871, P.R. China

{xhji, liuzimin, czhu, hzjia, xdxie, wgao}@jd1.ac.cn

Abstract. Integer Motion Estimation (IME) for block-based video coding introduces significant challenges in power consumption and silicon area usage with the adoption of more complex coding tools and higher resolution. To conquer these problems, this paper proposes an Binary Adaptive Luminance Mapping (BALM) algorithm by exploiting the local correlation in image and give a Very-large-scale Integration (VLSI) architecture for its implementation. We test the algorithm performance with different Number of Truncated Bits (NTB). And, the experimental results show that our proposed BALM achieves higher rate-distortion (RD) performance compared with previous proposed bit reduction approach and PSNR degradation is relatively small when $NTB \leq 5$ using our scheme. And, the NTB4 BALM can achieve 37.3% silicon area saving and power consumption reduction with just PSNR loss of 0.1 dB in our proposed IME architecture.

Keywords: integer motion estimation, binary adaptive luminance mapping, VLSI, video coding.

1 Introduction

The H.264/AVC video coding standard [1][2] can save 25%-45% and 50%-70% of bit rate while maintaining the same video quality compared with MPEG-4 advanced simple profile and MPEG-2, respectively [3][4]. The high coding efficiency comes from the new features of the standard such as variable block sizes (VBS) and multiple reference frames (MRF) [2] in Integer Motion Estimation (IME). However, these new techniques also increase the computation complexity and power consumption dramatically in a video encoder.

Among all the signal processing techniques required for video compression, IME is the most computational intensive and hence consumes the largest amount of hardware resource and power. Therefore, low complexity motion estimation that reduces the computation complexity and power consumption with reasonable rate-distortion (RD) performance are required, especially for Very-large-scale Integration (VLSI)-based implementations.

There are many techniques to reduce the complexity and the power consumption of the IME. Bit reduction is frequently used optimization technique. Bit Truncation (BT)

[5][6][7] that drops some least significant bits of 8-bit pixels in Sum-of-Absolute-Difference (SAD) calculation (used in block matching) is one type of bit reduction method. BT includes both the fixed bit truncation [5] and the adaptive bit truncation [6][7], but in all these methods, the least significant bits are truncated directly for each pixel. The number of bits that can be truncated is very limited and it usually causes higher performance degradation with the increase of Number-of-Truncated-Bits (NTB). Recently some novel bit reduction methods [8]-[11] are proposed for ME hardware implementation. In [8], current and reference frames are filtered with a multi band-pass filter. Then the filtered frame is compared with original frame to get the one bit-depth frame. Finally, low cost one bit matching criterion using Exclusive-OR (EX-OR) operations is adopted in the ME searching. In [9], the multi band-pass filter adopted in [8] is modified to avoid multiplication operations. In [10], a two bit-depth frame representation method is presented to improve the accuracy of motion estimation. However, excess computation is brought by low bit-depth generation method. A two bits ME using constrained one-bit transform is proposed in [11]. It employs simple operations to improve performance of one bit ME. And, it can generally provides better motion estimation performance compared to two bit-depth based ME. In this paper, we will introduce a novel bit reduction method by using locality property of an image which can be easily applied to VLSI-based ME architecture for silicon area reduction and power saving with less performance deterioration.

The remainder of this paper is organized as follows. Section 2 gives an introduction to the proposed Binary Adaptive Luminance Mapping algorithm. In Section 3, Binary Adaptive Luminance Mapping architecture and its application on a typical motion estimation are presented. Performance analysis as well as the comparisons with other works are discussed in Section 4. Finally, some concluding remarks are drawn in Section 5.

2 Adaptive Binary Luminance Mapping Algorithm

In this section, a novel adaptive binary luminance mapping algorithm is presented. By the nature of a typical video sequence, the spatially adjacent luminance data are highly correlated [5]. Generally, luminance pixels in a video are represented in 8~12 bits. The most significant bits are typically important in describing the skeleton of the image while the least significant bits play roles in describing the details. Intuitively, dropping some least significant bits of 8-bit luminance pixels during block matching using SAD calculation may not affect the ME performance. Reducing the bits of operands in SAD calculation can decrease the width of data path logic of a ME search engine which translates into less gate count, smaller routing area and power consumption. Less gate count and smaller routing area can greatly reduce challenges in the chip back-end design.

Bit Truncation (BT) [5][6][7] in which the least significant bits are truncated directly for each luminance pixel in an image is the common adopted algorithm for ME bit reduction. The N-bit BT means luminance dynamic range is reduced to 2^{8-N} levels in which the N least significant bits of pixels are dropped. By using BT, full

luminance dynamic range (typically from 0 to 255) is directly mapped to the low dynamic range from 0 to $2^{(8-N)} - 1$. The problem of this method is that the number of bits that can be reduced is very limited and it usually causes higher performance degradation when large NTB is used.

Towards conquering the problem, our proposed new algorithm is based on the observation that the luminance values in a certain area (a 8x8 block, for example) have a low relative dynamic range according to the local luminance correlation of an image. It is also noted that the block-based motion estimation uses a block as a basic unit for matching. By taking advantage of these observations, a more precise algorithm can be proposed by using adaptive luminance mapping for each block, as illustrated in Fig.1. This implies that different block may have different dynamic range centered at a different median value and hence the pixels values in a search window (reference picture) have to be adaptively mapped to this range before matching is performed.

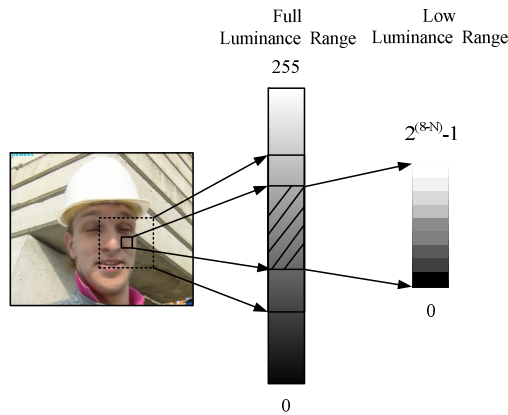


Fig. 1. Motivation of Adaptive Luminance Mapping Methods

The proposed Adaptive Luminance Mapping (ALM) is also illustrated in Fig.2.

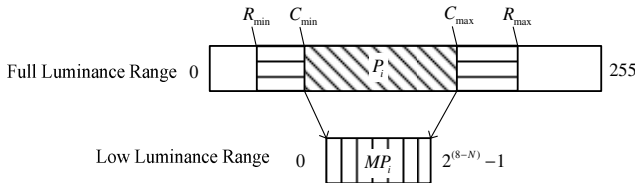


Fig. 2. Adaptive Luminance Mapping

Here C_{\max} and C_{\min} are respectively the minimum and maximum luminance values in the current block. And R_{\max} and R_{\min} are the maximum and minimum luminance values in the reference window. P_i is an arbitrary pixel in the current block (or in the reference window). MP_i denotes the luminance value of P_i after applying the proposed mapping algorithm. N is the number of reduced bits. The ALM can be defined by:

$$MP_i = \begin{cases} 0 & P_i < C_{\min} \\ \frac{P_i - C_{\min}}{C_{\max} - C_{\min} + 1} \times 2^{(8-N)} & C_{\min} \leq P_i \leq C_{\max} \\ 2^{(8-N)} - 1 & P_i > C_{\max} \end{cases} \quad (1)$$

To simplify the mapping for hardware implementation, some adjustments are employed for the mapping algorithm, as is shown in Fig.3.

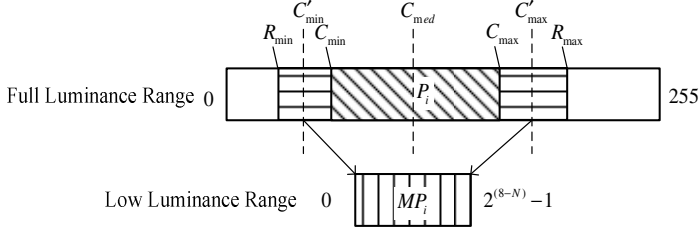
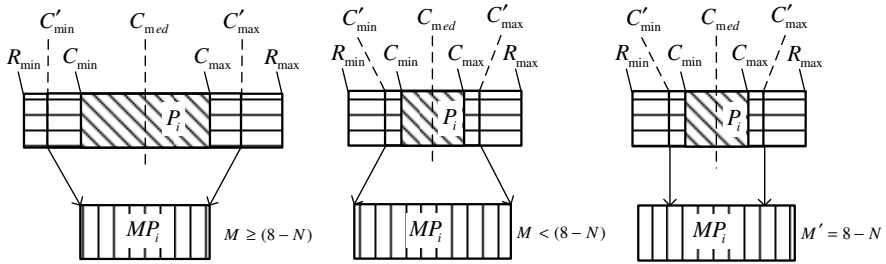


Fig. 3. Binary Adaptive Luminance Mapping

Here, $C_{med} = \frac{C_{\min} + C_{\max}}{2}$. C_{\min} and C_{\max} are respectively extended to C'_{\min} and C'_{\max} ($C'_{\min} \leq C_{\min}$, $C'_{\max} \geq C_{\max}$) which is also centered at C_{med} to satisfy the following constraint (rounding to the nearest integer bit).

$$\begin{cases} 2^{(M-1)} \leq C_{\max} - C_{\min} + 1 < 2^M \\ C'_{\max} - C'_{\min} + 1 = 2^M \end{cases} \quad (2)$$

M is the number of bits used to represent the difference between C'_{\min} and C'_{\max} . We can get two kinds of ALM as shown in Fig.4 (a) and Fig.4 (b) based on the value of M .



(a) ALM for $M \geq (8 - N)$ (b) ALM for $M < (8 - N)$ (c) Revised ALM for $M < (8 - N)$

Fig. 4. Two kinds of ALM

As P_i is expressed in integer values, it is unnecessary to perform the mapping which will also deteriorate the distortion for pixels in reference window as shown in Fig.4 (b). To minimize the distortion for reference pixels, M is revised to M' by Eq.(3).

$$M' = \text{Max}(M, 8 - N) \quad (3)$$

Fig.4 (c) shows the revised ALM using M' when M is less than $8 - N$.

And, C'_{\max} and C'_{\min} are updated using the revised M' . Hence, the ALM can be described as follows:

$$MP_i = \begin{cases} 0 & P_i < C'_{\min} \\ \left\lfloor \frac{P_i - C'_{\min}}{C'_{\max} - C'_{\min} + 1} \times 2^{(8-N)} \right\rfloor & C'_{\min} \leq P_i \leq C'_{\max} \\ 2^{(8-N)} - 1 & P_i > C'_{\max} \end{cases} \quad (4)$$

So we can get our scheme, Binary Adaptive Luminance Mapping (BALM) using only shift and subtract operation:

$$BMP_i = \begin{cases} 0 & P_i < C'_{\min} \\ (P_i - C'_{\min}) >> K & C'_{\min} \leq P_i \leq C'_{\max} \\ 2^{(8-N)} - 1 & P_i > C'_{\max} \end{cases} \quad (5)$$

Here, $K = M' - (8 - N)$ and it is a constant value for each current block. BMP_i is the luminance pixel value after the BALM. The sum of absolute differences (SAD) for the $N \times N$ blocks can be defined by:

$$SAD = \sum_{i=0}^N \sum_{j=0}^N |CMP_{ij} - RMP_{ij}| \quad (6)$$

Here, CMP_{ij} and RMP_{ij} are BALM pixels in the current and reference blocks respectively. Finally, we can get the Rate-Distortion cost function for the reduced-bit motion estimation using BALM:

$$COST = SAD \ll K + \lambda \times MVD_COST \quad (7)$$

3 The VLSI Architecture

The proposed BALM VLSI structure includes two types of mapping unit: binary adaptive luminance mapping unit (MU) and simplified binary adaptive luminance mapping unit (SMU). Fig.5 (a) presents the VLSI implementation of the BALM (Eq.(5)). Here, $MP_{\max} = 2^{(8-N)} - 1$. Reference pixels in the search window can be mapped by MU. When BALM is performed on the original pixels, the original P_i always satisfy the constraint of $C'_{\min} \leq P_i \leq C'_{\max}$. Hence, the compare operation in Eq.(5) can be simplified for the mapping of original pixels. And, a simplified VLSI architecture (SMU) including only one subtracter and one shift as shown in Fig.5 (b) is proposed for the original pixels.

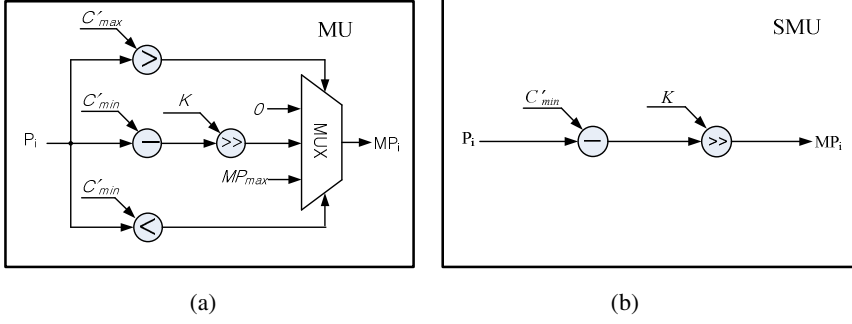


Fig. 5. (a) binary adaptive luminance mapping unit (b) Simplified binary adaptive luminance mapping unit

Fig.6 shows the architecture diagram of a typical ME engine adopting the proposed BALM for silicon area and power reduction. The binary mapping parameters generation module (BALM Gen) generates the C'_{min} , C'_{max} , K and MP_{max} after the fetch operation of current micro-block. Pixels in the current micro-block are mapped by SMU Array and sent to the current buffer at the beginning of search. Reference pixels read from on-chip SRAMs are mapped by MU Array before shifted into Reference Buffer. And, the Rate-Distortion cost is calculated in the Processing Element Array (PEA) for each search.

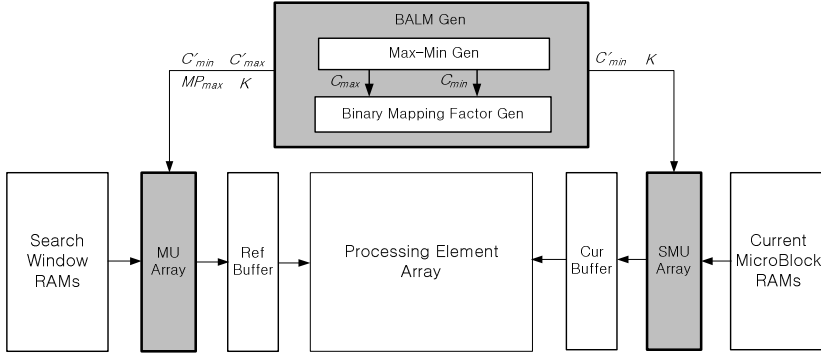


Fig. 6. Typical ME Engine Adopting the Proposed BALM

4 Experimental Results

In this section, we will give out the experimental results of the proposed BLAM and the comparisons with other works. Fig.7 shows the comparison between the ME reference window for BT [5][6][7] and the proposed BALM algorithms. The best matching positions for the current block in the reference window are marked as red block. The BT and BALM error maps contribute to the luminance distortion of the reference window for BT and BALM algorithms, respectively. It can be seen that luminance

distortion for the BT is quite irregular and the best matching position suffers from large luminance distortion which will greatly deteriorate the ME searching performance. The BALM behaves a more ‘smart’ distortion pattern. Reference region in which pixel is similar to the luminance value in the current block will get small distortion. And, large distortion in the BALM usually occurs at the most unlikely matching position.

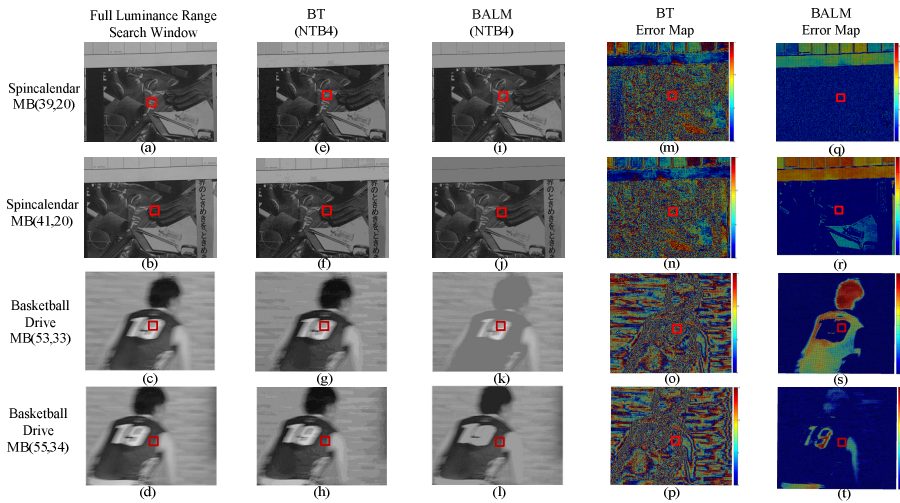


Fig. 7. Reference Window Samples of BT and BALM. (a) - (d) The full luminance range reference window for different micro-block. (e) - (h) The NTB4 reference window using BT algorithm. (i) - (l) The NTB4 reference window using proposed BALM algorithm. (m) - (p) The NTB4 reference window error map using BT algorithm. (q) - (t) The NTB4 reference window error map using BALM algorithm.

Four 720P and 1080P video sequences *City*, *Cyclists*, *Kimono* and *Spincalendar* are used for the RD performance evaluation. Fig.8 shows the simulation results of BT and our proposed BALM on Rate-Distortion for the four video sequences. And different NTBs are chosen for the performance evaluation. And the performance comparison between previous proposed bit reduction MF1BT[9] and C1BT[11] approaches is shown in Fig.9.

Contrary to the BT approach, our proposed BALM achieves higher RD performance. The PSNR degradation is relatively small when $NTB \leq 5$ using our scheme. For the 1 bit-depth and 2 bit-depth ($NTB = 7$ and $NTB = 6$) ME, the proposed BALM performs better than the previous MF1BT and C1BT approach. And, The VLSI implementation of BALM is as easy as the implementation of BT approach as result of simple operation. Finally, both BT and BALM are implemented in our proposed IME architecture [12]. We implemented different bit truncation schemes (for comparison purpose) on the proposed IME architecture. The silicon areas of different bit truncation schemes are measured by gate counts. The results are given in TABLE I. The NTB4 BALM can achieve 37.3% silicon area saving which can translate into proportionate reduction in power consumption of IME with just PSNR loss of 0.1 dB. And, it is finally adopted in our proposed IME architecture.

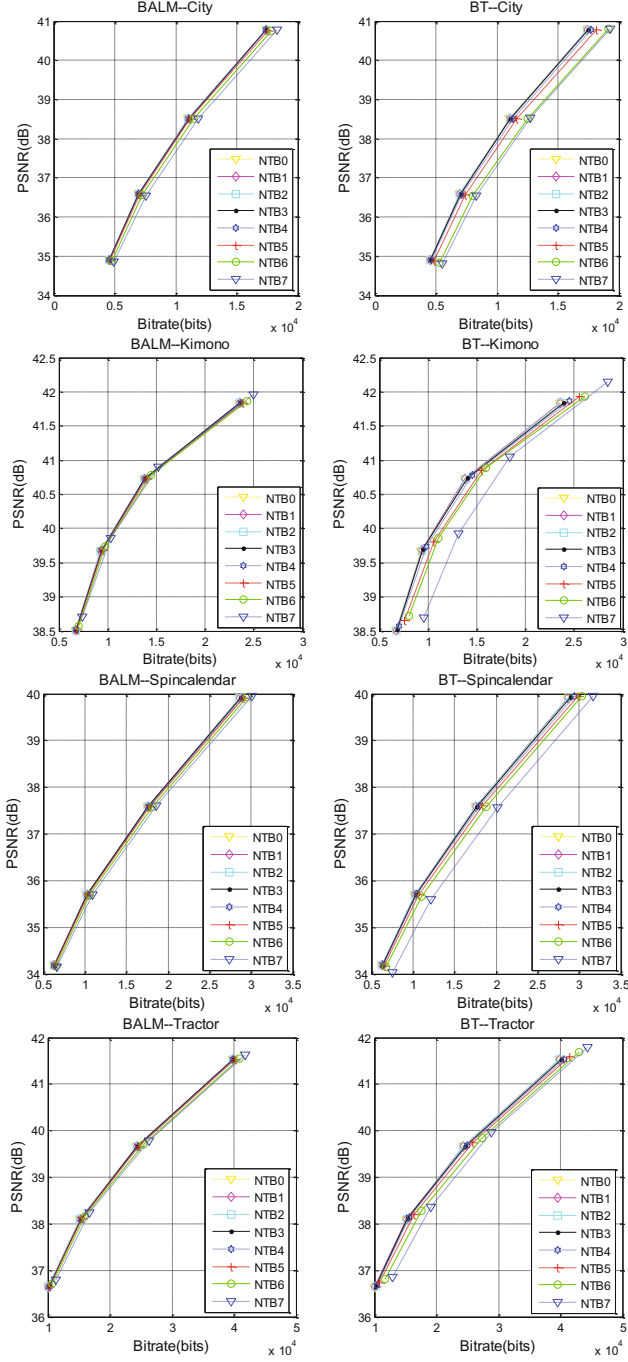


Fig. 8. Simulation Results for BALM and BT, With Hierarchical Search (FME ON) AND Search Range of 128x64 Pixels

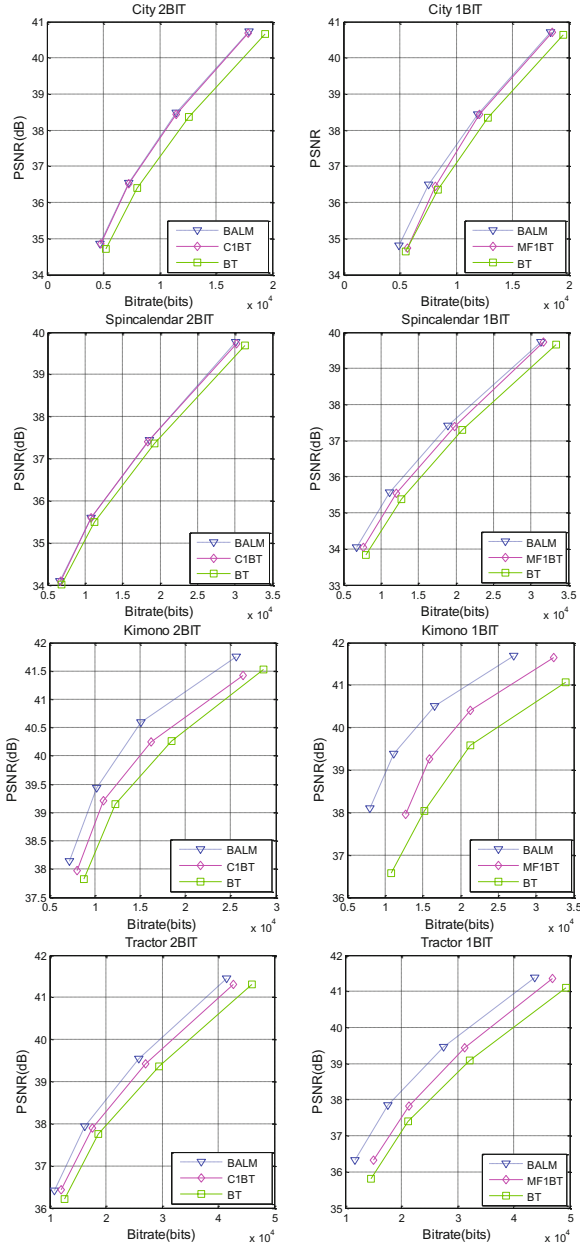


Fig. 9. Simulation Results for BALM, MF1BT and C1BT, With Hierarchical Search (FME OFF) AND Search Range of 128×64 Pixels

Table 1. Hardware Implementation Results Of Different Bit Reduction Schemes Results

NTB	BALM		BT	
	Gate Count (K)	Gate Count Saving	Gate Count (K)	Gate Count Saving
0	260	0.0%	260	0.0%
1	233	10.2%	227	12.5%
2	216	16.9%	209	19.5%
3	190	26.9%	178	31.7%
4	163	37.3%	141	45.7%
5	139	46.6%	113	56.5%
6	119	54.1%	97	62.6%

5 Conclusion

In this paper, we proposed a Binary Adaptive Luminance Mapping algorithm by using a dynamic mapping for each block which is based on the local pixel correlation of an image and give an architecture for its VLSI implementation. The proposed BALM VLSI architecture can be implemented using only small amount of integer arithmetic subtracters and shifts for silicon area and power reduction. Experimental results show that our proposed BALM achieves higher RD performance compared with previous algorithms and PSNR degradation is relatively small when $NTB \leq 5$ using our scheme. And, the NTB4 BALM can achieve 37.3% silicon area saving and reduction in power consumption with just PSNR loss of 0.1 dB in our proposed IME architecture.

References

1. Draft ITU-T Rec. Final Draft Int. Standard of Joint Video Specification, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC (May 2003)
2. Wiegand, T., Sullivan, G.J., Bjøntegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* 13(7), 560–576 (2003)
3. Iain, E.G., Richardson, H.: 264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia. ch. 7, pp. 225–267. Wiley, New York (2003)
4. Wiegand, T., Schwarz, F.K.H., Joch, A., Sullivan, G.: Rate constrained coder control and comparison of video coding standards. *IEEE Trans. Circuits Syst. Video Technol.* 13(7), 688–703 (2003)
5. Moshnyaga, V.G.: An MSB truncation scheme for low-power video processors. In: *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, pp. 291–294 (1999)
6. He, Z.-L., Tsui, C.-Y., Chan, K.-K., Liou, M.L.: Low-Power VLSI Design for Motion Estimation Using Adaptive Pixel Truncation. *IEEE Trans. Circuits Syst. Video Technol.* 10(5), 669–678 (2000)
7. Bahari, A., Arslan, T., Erdogan, A.T.: Low power variable block size motion estimation using pixel truncation. In: *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 3663–3666 (2007)

8. Natarajan, B., Bhaskaran, V., Konstantinides, K.: Low-complexity block-based motion estimation via one-bit transforms. *IEEE Trans. Circuits Syst. Video Technol.* 7(4), 702–706 (1997)
9. Ertürk, S.: Multiplication-free one-bit transform for low-complexity block-based motion estimation. *IEEE Signal Process. Lett.* 14(2), 109–112 (2007)
10. Kim, N.J., Ertürk, S., Lee, H.Y.: Two-Bit Transform Based Block Motion Estimation using Second Derivatives. *IEEE Trans. Consumer Electron* 55(2), 902–910 (2009)
11. Urhan, O., Ertürk, S.: Constrained one-bit transform for low complexity block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.* 17(4), 478–482 (2007)
12. Yin, H.B., Jia, H.Z., Qi, H.G., Ji, X.H., Xie, X.D., Gao, W.: A Hardware-Efficient Multi-Resolution Block Matching Algorithm and Its VLSI Architecture for High Definition MPEG-Like Video Encoders. *IEEE Trans. Circuits Syst. Video Technol.* 20(9), 1242–1254 (2010)