Efficient Hardware Implementations of Low Bit Depth Motion Estimation Algorithms

Anıl Çelebi, *Student Member, IEEE*, Oğuzhan Urhan, *Member, IEEE*, İlker Hamzaoğlu, *Member, IEEE*, and Sarp Ertürk, *Member, IEEE*

***Abstract*—In this paper, we present efficient hardware implementation of multiplication free one-bit transform (MF1BT) based and constraint one-bit transform (C-1BT) based motion estimation (ME) algorithms, in order to provide low bit-depth representation based full search block ME hardware for real-time video encoding. We used a source pixel based linear array (SPBLA) hardware architecture for low bit depth ME for the first time in the literature. The proposed SPBLA based implementation results in a genuine data flow scheme which significantly reduces the number of data reads from the current block memory, which in turn reduces the power consumption by at least 50% compared to conventional 1BT based ME hardware architecture presented in the literature. Because of the binary nature of low bit-depth ME algorithms, their hardware architectures are more efficient than existing 8 bits/pixel representation based ME architectures.**

***Index Terms*—Low bit-depth motion estimation, motion estimation hardware, source pixel based linear arrays.**

I. Introduction

BLOCK motion estimation with Sum of Absolute Differences (SAD) matching of pixels has been adopted as standard approach for motion estimation (ME) in video encoders. The one-bit transform (1BT) has been proposed in [1] to reduce the computational complexity of the matching process in ME by transforming video frames into 1 bit/pixel representations and performing ME using these binary representations. In [2], a new binarization kernel is proposed for 1BT to facilitate a multiplication free transform for reduced transform complexity, referred to as multiplication free one-bit transform (MF1BT). An early termination scheme for binary ME is presented in [3]. In [4], a two-bit transform (2BT) is proposed to improve ME accuracy compared to 1BT by constructing two bit-planes for each video frame and performing ME using the 2BT representations. Recently, the constrained one-bit transform (C-1BT) is presented in [5] and it is shown that the C-1BT can provide increased ME accuracy compared to 2BT at much lower complexity.

Manuscript received October 30, 2008; revised February 09, 2009. Current version published April 24, 2009. This work was supported by the Scientific and Technical Research Council of Turkey (TUBYTAK) under Contract 107E179, and also by the Turkish State Planning Organization Project DPT 2008K-120800. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Roman Genov.

A. Çelebi, O. Urhan, and S. Ertürk are with Kocaeli University Laboratory of Image and Signal Processing (KULIS), Department of Electronics and Telecommunication Engineering, University of Kocaeli, 41040 Kocaeli, Turkey (e-mail: anilcelebi@kou.edu.tr; urhano@ieee.org; sarp@ieee.org)

İ. Hamzaoğlu is with the Faculty of Engineering and Natural Sciences, Sabanci University, 34956 Istanbul, Turkey (hamzaoglu@sabanciuniv.edu)

Digital Object Identifier 10.1109/LSP.2009.2017222

In this paper, we present efficient hardware implementations of MF1BT and C-1BT based ME for the first time in the literature, in order to provide low cost ME hardware for real-time video encoding. MF1BT based ME hardware can be used when the ME hardware cost is very important but a small loss in the performance can be tolerated, whereas C-1BT based ME hardware can be used when a small increase in the ME hardware cost can be tolerated but ME performance is very important.

Binary ME hardware implementations are presented in [1], [6], [7]. In [1], a motion vector based linear arrays (MVBLA) architecture is used for implementing 1BT based ME. The source pixel based linear arrays (SPBLA) architecture is introduced in [6], but it is not used for implementing low bit-depth ME approaches so far. We used SPBLA architecture for implementing low bit depth ME for the first time in the literature. The proposed SPBLA based implementation results in a genuine data flow scheme which significantly reduces the number of data reads from the current block memory, which in turn reduces the power consumption by at least 50% compared to MVBLA based 1BT ME hardware presented in [1].

A fast binary ME algorithm based on a binary pyramid structure and its hardware architecture are presented in [7], but this approach has a higher complexity compared to MF1BT and C-1BT based ME approaches because it uses three different bit planes which increases the memory bandwidth compared to proposed 1BT based hardware architecture. In [7], neither ASIC nor FPGA implementation results are presented and their 1-D systolic array implementation uses two times more processing elements (PE) than the proposed 1BT based hardware architecture.

Many hardware architectures for ME algorithms using 8 bits/pixel video frames are proposed in the literature and several examples can be found in [8]–[12]. In [8], a hardware architecture for hierarchical ME with 8 bits/pixel representation and an example FPGA implementation together with a hardware complexity analysis is provided. In [9], an efficient variable block size ME hardware for H.264 is presented. In [10], a high performance hardware architecture with a new data reuse method for the search window data is proposed and a systolic register array is utilized to reduce the data read count for the reference data. In [11], memory bandwidth efficient hardware architecture is proposed with a new data reuse scheme that considers integer and fractional motion estimation together. A reconfigurable VLSI architecture to efficiently utilize data reuse on the search window based on a "meander"-like scan approach is proposed in [12] where 30% lower on-chip memory access ratio is achieved.

Because of the binary nature of the low bit-depth ME algorithms their hardware architectures are more efficient than existing 8 bits/pixel representation based ME hardware architectures. ME hardware implementations based on 8 bits/pixel representations require 2-D PE systolic arrays to achieve real-time ME for high resolution video (such as 720 p@30 fps) while low bit-depth representation based ME hardware implementations can achieve real-time ME for high resolution video by using 1D PE systolic arrays.

II. MF1BT and C-1BT Based Block Motion Estimation

In [1], 1BT using a multi-band pass filter that has 25 nonzero elements is utilized to obtain filtered images. The filtered images are compared to the original image frames to create the one-bit images. In this case, non-integer operations are required for the normalization stage of the filtering which has comparatively higher computational complexity. In [2], a novel diamond shaped kernel which is called multiplication free is proposed to decrease the computational burden of the filtering stage of 1BT.

2BT improves ME accuracy compared to 1BT, and C-1BT can provide increased ME accuracy compared to 2BT at much lower complexity [5]. In C-1BT, in addition to the 1BT bit-plane, a constraint mask is used while evaluating a block match. The constraint mask is used to decide if pixels are reliable enough to be considered in the 1BT matching process. In C-1BT the standard bit-plane is obtained as in conventional 1BT in the form of

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

where B shows the 1BT bit-plane, I and I_F denote the original and multi-bandpass filtered video frames, respectively, and (i,j) shows the spatial pixel position. A constraint mask (CM) is introduced for this purpose as

$$CM(i,j) = \begin{cases} 1, & \text{if } |I(i,j) - I_F(i,j)| \geq D \\ 0, & \text{otherwise} \end{cases} . \tag{2}$$

The CM indicates whether a pixel is close to the 1BT threshold (which is actually the filtered version of the original frame) by a certain distance D, or not. The constrained number of non-matching points (CNNMP) criterion is then used to evaluate the match of two blocks as in (3), shown at the bottom of the page, where $||$, $\bullet$, and $\oplus$ denote binary OR, AND, and EXOR operations respectively. Note that in case of MF1BT, the number of non-matching points (*NNMP*) criterion is obtained by omitting the first part (CM influence) in *CNNMP*. The search location for which the smallest *NNMP or CNNMP* value is obtained is considered as the motion vector of the current block.

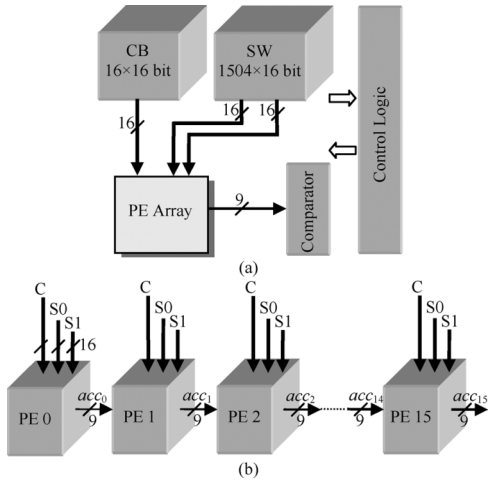

Fig. 1. (a) Proposed MF1BT based ME hardware and (b) PE Array block.

III. Proposed Hardware Architectures

Motion vector based linear arrays (MVBLA) and Source pixel based linear arrays (SPBLA) are commonly used hardware architectures for hardware implementations of ME algorithms [6]. In MVBLA, processing elements (PEs) in the systolic array compute the matching criterion for different search locations independently from each other. In SPBLA, PEs in the systolic array compute the matching criterion for each search location together. In SPBLA, there is a single interconnect between the PE array and the comparator, while, in MVBLA all of the PEs have connections to the comparator.

The proposed SPBLA based hardware architecture for MF1BT ME is shown in Fig. 1. The SPBLA based hardware architecture for C-1BT ME is the same except the extra memory blocks used for CMs. As shown in Fig. 1(a), there are two on chip memories in the architecture, one of them is used for storing the current block (CB) and the other is used for storing the search window (SW). For a block size of $M \times M$ and a search range of $[-m, n]$, an $M \times M$ bits memory is needed for CB and $M \times (1+m+n) \times (M+m+n)$ bits memory is needed for the SW.

The PE array block is shown in Fig. 1(b). It can be seen that the number of non-matching points accumulation is performed sequentially through the PEs in the PE array. Therefore, after the first 15 cycles, the matching criterion for one candidate location is computed in every clock cycle. Since there are 1024 candidate search locations in a search range of [−16, 15], 1039 clock cycles are needed to compute the motion vector for the current 16×16 block for a search range of [−16, 15].

$$CNNMP(m,n) = \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} \left\{ \left[CM^t(i,j) \,||\, CM^{t-1}(i+m, j+n) \right] \bullet \left[B^t(i,j) \oplus B^{t-1}(i+m, j+n) \right] \right\} \tag{3}$$

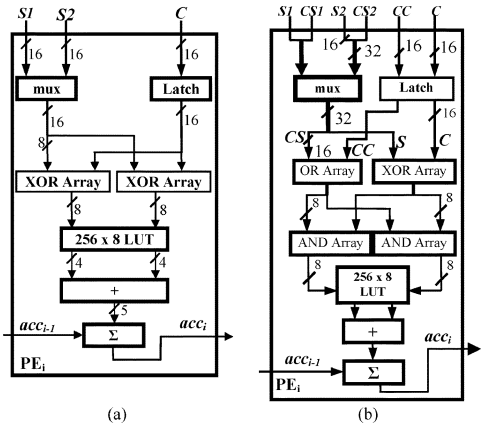

Fig. 2. (a) PE architecture for MF1BT ME hardware and (b) PE architecture for C-1BT based ME hardware.

The PE architectures for MF1BT and C-1BT based ME approaches are shown in Fig. 2(a) and (b), respectively. The PE architecture shown in Fig. 2(a) has two ports (S1, S2) for reading the reference block from the SW memory and one port (C) for reading the current block from the CB memory. To compute NNMP criterion for a candidate location, the pixels in the corresponding reference block and the pixels in the corresponding current block are XORed using *XOR* arrays and the number of ones output by the *XOR* arrays are counted using a dual input and dual output look up table (LUT) and a 4-bit adder. Finally, the accumulated partial NNMP count coming from the previous PE and the partial NNMP count of the current PE are added and the result is sent to the next PE.

C-1BT based ME hardware computes CNNMP criterion for a candidate location similar to the computation of NNMP criterion by MF1BT based ME hardware. However, since the CNNMP measure requires additional logic operations compared to the NNMP measure, the PE architecture for C-1BT ME hardware contains additional blocks such as the *AND* array and the *OR* array as shown in Fig. 2(b).

The data flow scheme of the proposed MF1BT based ME hardware is presented in Table I. In this table, $C0$ denotes the current block pixels in the first row of the current 16×16 block and $S0,0$ denotes the 16 reference block pixels in the (0,0) to (0,15) coordinates of the search window. In this table, terms shown in square brackets denote the pixels that are read from the *Latch* block instead of the on-chip memory, while terms not shown in square brackets denote the pixels that are read from the on-chip memory.

As shown in Table I, after the first 15 cycles, the NNMP measure for one candidate location is computed in every clock cycle. All of the current block pixels are latched in the first 15 clock cycles, and a memory read operation is not performed for the current block pixels after the 15th clock cycle. Therefore, only 16 memory reads are required from the CB memory. However, in the MVBLA based 1BT ME hardware presented in [1], 1024 memory reads are required from the CB memory for the same search range size of $[-16, 15]$. Therefore, using SPBLA hardware architecture for binary ME significantly reduces the number of data reads from the CB memory compared to the architecture presented in [1].

TABLE I
DATA-FLOW SCHEME OF THE MF1BT BASED ME HARDWARE

#Cycle	Input Data			Inputs of PEs					
	C	*S1*	*S2*	PE-0	PE-1	...	PE-14	PE-15	Output
0	C0	S0,0		C0-S0,0					
1	C1	S1,0		[C0]-S1,0	C1-S1,0				
2	C2	S2,0		[C0]-S2,0	[C1]-S2,0				
⋮	⋮	⋮		⋮	⋮				
14	C14	S14,0		[C0]-S14,0	[C1]-S14,0	...	C14-S14,0		
15*	C15	S15,0		[C0]-S15,0	[C1]-S15,0	...	[C14]-S15,0	C15-S15,0	D(0,0)
16	No read operations	S16,0		[C0]-S16,0	[C1]-S16,0	...	[C14]-S16,0	[C15]-S16,0	D(1,0)
⋮		⋮		⋮	⋮		⋮	⋮	⋮
31		S31,0		[C0]-S31,0	[C1]-S31,0	...	[C14]-S31,0	[C15]-S31-0	D(16,0)
32		S32,0	S0,1	[C0]-S0,1	[C1]-S32,0	...	[C14]-S32,0	[C15]-S32,0	D(17,0)
⋮		⋮	⋮	⋮	⋮		⋮	⋮	⋮
46		S46,0	S14,1	[C0]-S14,0	[C1]-S14,0	...	[C14]-S14,1	[C15]-S46,0	D(31,0)
47			S15,1	[C0]-S15,1	[C1]-S15,1	...	[C14]-S15,1	[C15]-S15,1	D(0,1)
*All of the PEs are functional									

TABLE II
SYNTHESIS RESULTS

ME Method	MF1BT		C-1BT	
Architecture	Proposed	[1]	Proposed	[1]
Frequency (MHz)	187	98	179	92
Number of LUTs Device Utilization	1121 (3%)	2541 (8%)	1721 (5%)	3082 (10%)
Block Rams	16	4	20	8
Dual Port Rams	16	16	32	32
ROMs	24	96	27	96

IV. IMPLEMENTATION RESULTS

The proposed ME hardware architectures are implemented in Verilog HDL and the implementations are verified with post P&R simulations using Mentor Graphics ModelSim.

The Verilog RTL implementations are synthesized to a Xilinx XC2VP30 FPGA using Synplify Premier tool. The synthesis results of the proposed ME hardware and the synthesis results of the MF1BT and C-1BT ME hardware implemented using the hardware architecture presented in [1] are presented in Table II. These results show that the area of the hardware architectures is similar but the performance of the proposed architecture is much better than the architecture presented in [1].

Power consumption analysis of the proposed architecture and the architecture presented in [1] are carried out using Xilinx XPower tool by using the switching activity of the hardware architectures obtained using Mentor Graphics ModelSim simulator as described in [13]. The power consumption analysis for several search positions in a sample video frame is given in Table III for C-1BT. These results show that the average power consumption of the architecture proposed in [1] is about 2 times higher than the proposed architecture, which clearly proves the efficiency of the proposed data flow scheme.

Both proposed MF1BT and C-1BT ME hardware architectures, when implemented on a Xilinx XC2VP30 FPGA, can perform 45 fps full-search ME for 720 p HDTV sized video frames for a search range of $[-16, 15]$ pixels. It is clear from Table II that the hardware complexity of the proposed architectures increases depending on the number of bit planes used in the ME method. Therefore, MF1BT based ME hardware can be used when the ME hardware cost is very important but a small loss

TABLE III
POWER CONSUMPTION RESULTS FOR C-1BT

Block Position	Motion Vector	[1](mW)	Proposed(mW)
(17,4)	(3,-6)	239	126
(17,5)	(5,-3)	304	120
(17,13)	(-9,0)	350	173
(21,2)	(3,-2)	250	112
(21,13)	(6,2)	326	146

TABLE IV
COMPARISON WITH 8 BITS/PIXEL ME HARDWARE

	[12]	Proposed MF1BT
Number of PEs	256	16
Search Range	[-16,16]	[-16,15]
Block Size	Variable	16×16
Cycle Count	1129	1039
Process	0.18μm	FPGA
Max Frequency	200MHz	196MHz
Power	423mW@180MHz	123mW@170MHz
Performance	720p@45fps	720p@45fps

in the performance can be tolerated. On the other hand, C-1BT based ME hardware can be used when a small increase in the ME hardware cost can be tolerated but ME performance is very important.

Comparison of the proposed ME hardware with the ASIC implementation of an 8 bits/pixel based ME hardware presented in [12] is shown in Table IV. Because of the binary nature of MF1BT ME algorithm, the proposed MF1BT ME hardware is more efficient than the 8 bits/pixel representation based ME hardware presented in [12] as it requires a much lower number of PEs. Furthermore the power consumption of the proposed approach is significantly lower. It is important to note that results provided for the proposed approach are for FPGA implementation while results in [12] are for ASIC implementation in 0.18 μm process. The power consumption of the proposed approach is expected to decrease even further in case of ASIC implementation.

ME hardware based on 8 bits/pixel representation require 2-D PE systolic array to achieve real-time ME for high resolution video while the proposed MF1BT ME hardware achieves real-time ME for high resolution video by using 1-D PE systolic array. Because each PE in 8 bits/pixel representation based ME hardware can process only one pixel in each clock cycle whereas each PE in MF1BT ME hardware can process 16 pixels in each cycle. In addition, in 8 bits/pixel representation based ME hardware, a 2-D parallel adder tree is used for adding the partial SADs computed by the PEs, whereas in the proposed MF1BT hardware the partial NNMP results computed by the PEs are added sequentially in each PE.

V. CONCLUSION

In this paper, efficient hardware implementation of MF1BT and C-1BT based ME algorithms are presented for the first time in the literature in order to provide low cost ME hardware for real-time video encoding. MF1BT ME hardware can be used when the ME hardware cost is very important but a small loss in the performance can be tolerated, whereas C-1BT based ME hardware can be used when a small increase in the ME hardware cost can be tolerated but ME performance is very important. In this paper, SPBLA hardware architecture is used for implementing low bit depth ME for the first time in the literature. The proposed SPBLA based implementation of low bit-depth ME results in a genuine data flow scheme which significantly reduces the number of data reads from the current block memory, which in turn reduced the power consumption by at least 50% compared to MVBLA based 1BT ME hardware presented in the literature. Because of their binary nature, low bit-depth ME hardware is more efficient than existing 8 bits/pixel representation based ME hardware.

ACKNOWLEDGMENT

The authors would like to thank Prof. Dr. G. Dündar from Bogazici University, Turkey, for his valuable comments. Some of the hardware and software tools used in this work were donated by Xilinx in scope of the Xilinx University Program.

REFERENCES

[1] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 702–706, Aug. 1997.
[2] S. Ertürk, "Multiplication-free one-bit transform for low-complexity block-based motion estimation," *IEEE Signal Process. Lett.*, vol. 14, no. 2, pp. 109–112, Feb. 2007.
[3] H. Lee and J. Jeong, "Early termination scheme for binary block motion estimation," *IEEE Trans. Consumer Electron.*, vol. 53, no. 4, pp. 1682–1686, Nov. 2007.
[4] A. Ertürk and S. Ertürk, "Two-bit transform for binary block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 938–946, Jul. 2005.
[5] O. Urhan and S. Ertürk, "Constrained one-bit transform for low complexity block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 478–482, Apr. 2007.
[6] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, 2nd ed. Norwell, MA: Kluwer , 1997.
[7] J.-H. Luo, C.-N. Wang, and T. Chiang, "A novel all-binary motion estimation (ABME) with optimized hardware architectures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 8, pp. 700–712, Aug. 2002.
[8] S. Yalcin, H. Ates, and I. Hamzaoglu, "A high performance hardware architecture for an SAD reuse based hierarchical motion estimation algorithm for H.264 video coding," in *Int. Conf. on Field Programmable Logic and Applications*, Aug. 2005, pp. 509–514.
[9] C. Chen, S.-Y. Chien, Y.-W. Huang, T.-C. Chen, T.-C. Wang, and L.-G. Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," *IEEE Trans. Circuits and Systems I*, vol. 53, no. 2, pp. 578–593, Feb. 2006.
[10] T.-C. Chen, Y.-H. Chen, S.-F. Tsai, S.-Y. Chien, and L.-G. Chen, "Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 568–577, May 2007.
[11] D. Ding, S. Yao, and L. Yu, "Memory bandwidth efficient hardware architecture for AVS encoder," *IEEE Trans. on Consumer Electron.*, vol. 54, no. 2, pp. 675–680, May 2008.
[12] C. Wei, H. Hui, T. Jiarong, and M. Hao, "A high-performance reconfigurable VLSI architecture for VBSME in H.264," *IEEE Trans. Consumer Electron.*, vol. 54, no. 3, pp. 1338–1345, Aug. 2008.
[13] M. Parlak and I. Hamzaoglu, "Low power H.264 deblocking filter hardware implementations," *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, May 2008.