Truncated Gray-Coded Bit-Plane Matching Based Motion Estimation and its Hardware Architecture

Anıl Çelebi, *Student Member*, IEEE, Orhan Akbulut, Oğuzhan Urhan, *Member*, IEEE,
Sarp Ertürk, *Member*, IEEE

Abstract — *This paper proposes an efficient low bit-depth representation based motion estimation approach which is particularly suitable for low-power consumer electronics devices. In the proposed approach motion estimation is carried out using bit truncated gray-coded image pixels. The corresponding hardware architecture is also designed and presented in this paper to show the effectiveness of the proposed approach. It is shown that the proposed approach provides improved motion estimation accuracy compared to conventional bit-truncation based approaches that are directly applied to binary coded pixel values. The proposed approach uses simple Gray-coding, that has very low-complexity and can be applied on a pixel-by-pixel basis. Hence, the comparatively more complex transformation processes required in One Bit-Transform or Two-Bit Transform based low bit-depth representation ME approaches are avoided. Experimental results show that the proposed approach also outperforms such low bit-depth representation based motion estimation methods previously presented in the literature, in terms of motion estimation accuracy*[1].

Index Terms — Motion estimation, gray-coding, bit truncation, hardware architecture, systolic arrays.

I. INTRODUCTION

Recent increases in computational processing capabilities of microprocessors as well as highly effective dedicated hardware implementations together with emerging video coding standards have increased video applications in consumer electronics devices. Moreover, widespread Internet access and increased capacity of storage media, have contributed to increases in video content distribution. However, mobile devices have typically limited processing capabilities and battery life, and therefore require low-complexity video compression techniques to enable efficient transmission of captured video over bandwidth limited channels.

Many consumer electronics devices, such as mobile phones and camcorders for example, typically use H.263 [1] or H.264/AVC [2] based techniques for video compression. Motion estimation (ME) is basically the most complex and processing power intensive part of the encoder. Although various low-complexity ME approaches have been presented in the literature to reduce the computational complexity of ME, only several efficient hardware implementations of such approaches have been proposed.

One-bit transform (1BT) based ME has been proposed in [3] as a low-complexity ME approach for video compression. In 1BT, video frames are converted into a single bit-plane by comparing them with their multi-band pass filtered version. In 1BT based ME, block matching is performed using Exclusive-OR (EX-OR) matching of bit-planes. ME using EX-OR matching of 1BTs enables efficient hardware implementation compared to conventional SAD (sum of absolute difference) matching, decreasing the computational load by nearly sixteen times at the expense of some accuracy loss.

A multiplication-free one-bit transform (MF1BT) is proposed in [4] to decrease the computational load of the 1BT presented in [3] by making use of a novel multiplication-free kernel to obtain bit-planes. In [5] a two-bit transform (2BT) based ME approach is introduced to improve the performance of 1BT based methods using an additional bit-plane, but thereby increasing computational load compared to 1BT. A constrained one-bit transform (C-1BT) based ME approach is proposed in [6] to introduce a constraint mask so as to use only 1BTs of reliable pixels in the matching process. The C-1BT based ME approach is shown to provide improved matching compared to other 1BT and 2BT based ME approaches.

Low bit-depth matching based ME approaches can also be combined with additional complexity reduction methods for further reduction in ME complexity. In [7], a partial distortion search approach combined with a sparse search point approach is utilized with C-1BT based ME to further reduce the computational load for software implementation, at the expense of a slight accuracy loss. An early termination scheme is combined with MF1BT based ME in [8], to further reduce computational complexity.

Various hardware implementations of ME algorithms are proposed in the literature. It is stated in [9] that one of the most important aspects influencing power consumption in ME hardware architectures is system memory bandwidth, and a higher system memory bandwidth increases power consumption. Most hardware architectures proposed for the block matching algorithm (BMA) in ME are designed using parallel architectures based on systolic or semi-systolic arrays [10-14].

In [10] a hardware architecture utilizing SAD reuse to reduce the system memory bandwidth for variable block size

[1] This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) and Korean Research Foundation (KRF) Cooperation Program project entitled "Low-Complexity Motion Estimation Techniques and Their System-on-Chip Implementation" (TUBITAK project number 107E179).

Authors 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@kocaeli.edu.tr, orhanakbulut@gmail.com, urhano@ieee.org, sarp@ieee.org).

Contributed Paper
Manuscript received June 28, 2009

motion estimation is proposed. An efficient hardware architecture for variable block size ME is obtained in [10] by removing the data dependency between the sub-partitions of macroblocks and modifying the prediction flow accordingly. In [11], a detailed architectural analysis of variable block size ME for H.264/AVC is investigated. Instead of using a 1D adder tree as in [10], a 2D adder tree architecture that increases parallelism and a new search scheme that improves data reuse is utilized in [11]. In [12] a hardware-oriented fast ME algorithm is proposed with the intra-/inter-candidate data reuse considerations. In [13], an adaptive search range algorithm is proposed for the software side and a SAD-tree based architecture is introduced in the hardware side for software/hardware co-solution to achieve high throughput ME for H.264/AVC HDTV. The common purpose of these hardware implementations is to reduce complexity and/or power consumption.

One way to reduce processing complexity and/or power consumption is to reduce the data amount to be processed. Therefore, architectures that make use of low bit-depth based representations such as 1BT and 2BT can therefore provide efficient hardware implementations. Several hardware implementations of binary ME approaches are presented in [3, 14-18]. In [3], all binary ME using 1BT and the hardware architecture based on a 1D linear PE (processing element) array is presented. An all binary hierarchical ME approach and the corresponding hardware design are presented in [14]. In [15], a platform based implementation of the approach proposed in [14] is presented with bus interlaced architecture. A fast binary ME algorithm for MPEG4 shape encoding is presented in [16] together with its hardware architecture. In [17], hardware architectures for 1BT based ME methods are proposed together with an efficient data flow scheme where the power consumption is reduced about 50% compared to the hardware architecture proposed in [3]. Recently, an extension of 1BT based ME hardware architecture presented in [17] to sub-pixel level is proposed in [18].

The number of arithmetic operations carried out in the PE array and adder structures is another aspect influencing the complexity and power consumption of ME hardware architectures. In the hardware architecture presented in [13], a 2D PE array composed of 256 PEs is used with an SAD tree and variable block size (VBS) adder tree, requiring 6368 full adders (FAs) in total. On the other hand, the 1BT based ME architecture presented in [17] requires only a total of 199 FAs.

Instead of using all 8-bits of pixel values for ME, it is proposed in [19] to use bit truncation by utilizing only a certain number of the most significant bits (MSB), by truncating the lower bits, in order to reduce the computational load. Alternatively, an adaptive, pixel bit-depth reduction technique based ME approach and its VLSI architecture is presented in [9], however additional processing is required to obtain the reduced bit-depth representations in this case compared to simple bit truncation. Bit truncation is furthermore applied to variable block size motion estimation in [20].

Bit truncation based ME hardware architectures are proposed in literature to reduce the computational complexity of 8-bits/pixel based BMA at the expense of some loss in ME accuracy [19, 20, 22]. In [19], the VLSI implementation of bit-truncation based ME is accomplished using a well known parallel architecture proposed in [21]. A variable length bit truncation technique based ME approach and its VLSI architecture is proposed in [22]. However, the hardware complexity of the architecture presented in [22] is relatively high compared to low bit-depth based ME architectures because the PE architecture proposed in [22] is designed to process both low bit-depth as well as full bit-depth pixels.

In [23], Gray-coded pixel values are used to obtain global motion in image sequences for video stabilization. This approach is employed for block matching based ME in [24] to investigate its possible utilization in ME for video coding.

This paper proposes to employ bit truncation on Gray-coded pixel values for low-complexity ME and it is shown that this approach provides improved prediction performance compared to existing low bit-depth representation based ME approaches such as 1BT, 2BT, MF1BT, and C-1BT. Furthermore, the proposed approach significantly reduces the binarization process which is comparatively more complex in 1BT based ME methods due to the filtering process. It is also shown that the proposed method outperforms conventional bit truncation based ME approaches in terms of ME accuracy. The proposed approach enables low-complexity and power efficient ME hardware architecture implementation.

II. Truncated Gray-Coded Bit-Plane Matching Based Motion Estimation

Gray-coding based block motion estimation is presented in [24] to reduce computational load of the motion estimation process particularly in hardware implementations. In this paper, it is proposed to use bit-truncation with Gray-coding to further reduce ME complexity and at the same time facilitate efficient hardware design.

It is possible to represent a pixel value that is quantized to 2^K grey quantization levels, and located at location (x,y) of frame f at time t in the form of

$$f^t(x,y) = a_{K-1}2^{K-1} + a_{K-2}2^{K-2} + \ldots + a_1 2^1 + a_0 2^0 \tag{1}$$

where a_k coefficients represent the natural binary code and take only binary values. If the k th bit-plane of frame t is represented as $b_k^t(x,y)$, it contains all a_k bits of level k. If a bit-depth of 8-bits/pixel is used, K is 8, and $b_0^t(x,y)$ is the least significant bit-plane, while $b_7^t(x,y)$ is the most significant bit-plane.

The gray-coded version of a pixel value can be computed from its natural binary codes as:

$$\begin{aligned} g_{K-1} &= a_{K-1} \\ g_k &= a_k \oplus a_{k+1} \end{aligned} \quad , \quad 0 \le k \le K-2 \tag{2}$$

where $\oplus$ shows the EX-OR operation. Since the gray codes of adjacent grey levels differ only in a single bit, it is more appropriate to use Gray-coded pixel values in EX-OR matching based ME.

In BM, the original block of the current frame is searched for, inside a search window in the reference frame (which is usually the previous frame), using a certain similarity measure. In Gray-coded bit-plane matching based ME the similarity between the current block of size $N \times N$ pixels located in frame t and the reference block located in frame $t-1$ can be calculated using a correlation measure (CM_{GC}) which is defined as

$$CM_{GC}(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{K-1} 2^k \left\{ g_k^t(i,j) \oplus g_k^{t-1}(i+m, j+n) \right\} \tag{3}$$
$$-s \le m,n \le s-1$$

where (m,n) and s denote the candidate displacement and search range, respectively. The displacement resulting in the lowest CM_{GC} value is assigned as the motion vector of the current block. Note that, a scaling factor of 2^k is utilized to include the weight of level k when computing the similarity measure, so that higher order bit-planes have higher weight.

The proposed truncated Gray-coding based bit-plane matching approach does not use all K bit-planes, but it makes only use of the highest M bit-planes to compute the similarity measure. If the number of truncated bits is shown as NTB, then the highest $M = K - NTB$ bit-planes are used in the matching process. Thus, the new correlation metric CM_{TGC} is defined as

$$CM_{TGC}(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=NTB}^{K-1} 2^{k-NTB} \left\{ \begin{array}{l} g_k^t(i,j) \\ \oplus g_k^{t-1}(i+m, j+n) \end{array} \right\} \tag{4}$$
$$-s \le m,n \le s-1$$

Note that basically the lower boundary of one of the summations shown in (3) is changed in the similarity measure computation of the proposed approach, but this provides an important reduction in computations. It is also obvious that this measure can be computed with only binary operations since the multiplication in (4) can be carried out using shift operations as it is power of 2.

The ME performance of the proposed method for different NTB values is provided in the experimental results section. Experimental results show that $NTB = 5$ gives the best performance in terms of motion estimation accuracy taking at the same time complexity into account; and only the most significant three Gray-coded bit-planes are utilized for ME in this case. The hardware design is therefore carried out for $NTB = 5$.

III. Hardware Design

Because of the binary nature of the proposed algorithm a 1D systolic array architecture is sufficient to provide real-time processing performance. The architectures proposed for 8-bits/pixel representation based ME algorithms mostly utilize 2D systolic arrays to process video in real time. A similar data throughput to 8 bit/pixel representation based 2D systolic array architecture is achievable by utilizing only a 1D systolic array for a binary ME method, since each PE in a binary ME hardware architecture can process multiple pixels at each clock cycle [3].

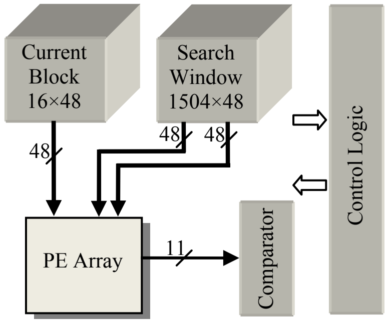

Fig. 1. Hardware architecture of proposed approach

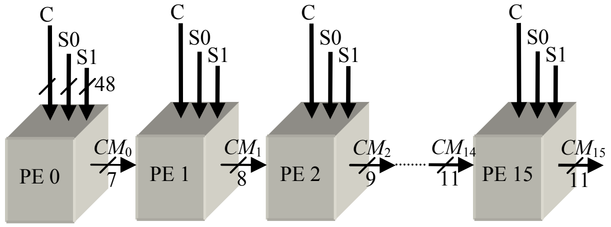

Fig. 2. PE Array

The overall hardware architecture proposed in this paper is shown in Fig. 1. The implemented architecture is capable of performing ME at macroblock level (16×16 pixels) for a search range of $[-16,15]$ pixels. A RAM block, of size 48-bit wide and 16 rows deep, is used to store the current macroblock (The macroblock size is 16×16 pixels and the most-significant 3 bits of the gray-coded bit-planes are used in the matching process). A dual port RAM block of size 48-bit wide and 1504 $47(for\_row\_scan) \times 32(for\_column\_scan)$ rows deep is used to store the search window. Let $S_{i,j}$ denote the 48-bit wide pixel vector that is located between the columns *j* and (*j*+47)th on the row *i* of the search window. Two different row vectors of the search window are needed during the operation of the proposed hardware architecture. Thus, a dual port memory is used to store the search window pixels.

ME at macroblock level with a search range of $[-16,15]$ pixels, requires a search window of size 47×47 pixels. Therefore, the theoretical minimum memory size required for the search window memory of the proposed ME architecture is actually 3-bits/pixel×47×47= 6,627k bits. This leads to a total minimum on-chip memory amount of 7,395k bits, including the current block memory, in theory. However, in the implementation, the total memory used for the proposed architecture is actually (1504×48) + (16×48) = 72,96k bits. This is much higher than the theoretical minimum, and it is actually possible to reduce this size by designing additional data scheduling hardware, however, this is out of the scope of this paper. For comparative evaluation it is useful to note that the total on-chip memory amount is 24,32k bits in [17] for 1BT based BMA hardware where a search window memory

TABLE I. DATA FLOW SCHEME OF PROPOSED ARCHITECTURE

Cycle Number	Input Data			Inputs of PEs					LOCATION OF BLOCK DISTIRTION THAT IS CALCULATED
	C	*S1*	*S2*	PE-0	PE-1	. . .	PE-14	PE-15	
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	CM(0,0)
16	C0	S16,0		[C0]-S16,0	[C1]-S16,0	…	[C14]-S16,0	[C15]-S16,0	CM(1,0)
.	.	.		.	.	.	.	.	.
.	.	.		.	.	.	.	.	.
.	.	.		.	.	.	.	.	.
31	C15	S31,0		[C0]-S31,0	[C1]-S31,0	…	[C14]-S31,0	[C15]-S31-0	CM(16,0)
32	C0	S32,0	S0,3	[C0]-S0,3	[C1]-S32,0	…	[C14],S32,0	[C15]-S32,0	CM(17,0)
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
46	C14	S46,0	S14,3	[C0]-S14,3	[C1]-S14,3	…	[C14]-S14,3	[C15]-S46,0	CM(31,0)
47	C15		S15,3	[C0]-S15,3	[C1]-S15,3	…	[C14]-S15,3	[C15]-S15,3	CM(0,1)
***All of the PEs are functional									

bigger than the theoretical minimum is used because of the abovementioned reasons. On the other hand, the on chip memory used in [11] is 208k bits for ME hardware with 8 bits/pixel representations.

The PE array consists of 16 PEs as shown in Fig. 2. All PEs in the PE array are basically identical to each other, and the only difference is the width of the *CM* input/output of each PE. The width of the *CM* output of each PE is determined by the range of possible values the *CM* output can take. In other words, it is determined by the maximum possible value of the adder tree and the *CM* input, which are summed to obtain the *CM* output. The output of the adder tree is fixed and 7-bit wide. The *CM* output of the first PE (CM_0) is therefore also 7-bit wide. The *CM* output of the next PE (CM_1) on the other hand needs to be 8-bit wide as it is the sum of two 7 bit values, i.e. the sum of the *CM* value of the previous PE and the output of its own adder tree. Considering the maximum possible values at each stage, the final *CM* output (CM_{15}) is obtained to be 11-bit wide. The PE architecture of the proposed hardware is shown in Fig. 3. In addition to *CM* inputs and outputs, there are three data inputs to each *PE*: one for the current macroblock (C) and two for the search window data (S1, S2). Three LUTs are used to obtain the EX-OR matching result, one for each gray-coded bit-plane, because three MSBs of gray-coded pixel values are used in the matching process. Note that the current macroblock data is not shifted through the PEs, but each PE uses the corresponding row of the current macroblock to compute the *CM* value of that row, so that actually a shared computation scheme is utilized. This is also seen in the data flow scheme of the proposed architecture shown in Table I. The multiplexer (Mux) block in the PE architecture selects the correct search data, depending on the control signal coming from the control block, according to the data flow scheme shown in Table I. The outputs of the Latch and the Mux blocks are inputted into the EX-OR array to compute their EX-OR distance. The output of the EX-OR array is separated into three parts, corresponding to the three separate bit-planes, according to the significance (weight) of each bit-plane. Because the most significant three bit-planes are used for each gray-coded pixel value, three 16-bit wide vector groups are constructed, and these groups are then applied to the inputs of the 3 LUTs. The PE architecture shown in Fig. 3 contains LUTs with two 8-bit wide inputs and two 4-bit wide outputs. The reason for this is to simplify the calculation of the number of ones in a 16-bit wide vector output of the EX-OR matching operation. For a 16-bit wide input, the depth of the LUT is needed to be 2^{16} whereas this number is 2^8, for 8-bit wide vectors; and thus, two smaller LUTs are sufficient to perform this operation with an additional 4-bit adder. The outputs of the three LUTs are then summed in the adder tree taking into account their corresponding weights. The output of the adder tree is 7 bits, so as to accommodate the largest possible value. Finally, the output of the adder tree is added to the *CM* output of the previous PE and this sum is forwarded to the *CM* input of the next PE.

Note that in Table I, square brackets show that the corresponding data is read from the latch instead of the current macroblock RAM. Thus, for each PE only a single memory read operation is needed for the current macroblock in the entire motion vector computation stage of a macroblock.

In Table I, C*i* represents the 48-bit wide vector located in the *i*th row of the current macroblock comprising the three MSBs of Gray-coded values of the 16 pixels in the *i*th row. S*i*,*j* represents the 48-bit wide vector located in the *i*th row between columns *j* and *(j+47)* in the search window. For example S0,0 denotes the three MSBs of the 16 pixels in the 0th row concatenated in the form of {(S0,0-2),(S0,3-5),…,(S0,45-47)}. Each PE in the PE array shown in Fig. 2 can process 16 pixels in one clock cycle so that 16×16 pixels can be processed in each cycle when all of the PEs in the array are utilized. As shown in Table I, 15 clock cycles are needed for the PE array to become fully functional.

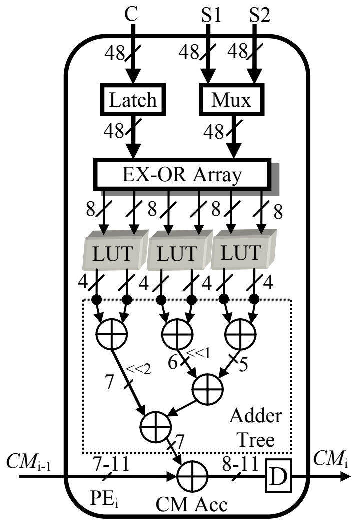

Fig. 3. PE Architecture of the proposed hardware

The hardware needs 1024 clock cycles for the computation of the motion vector for one macroblock, in addition to this 15 clock cycle offset. More detailed information about this data flow scheme, typical to 1D systolic array architectures, is available in [25].

The total full adder amount needed for the PE array of the proposed architecture is 523, where 16×23=368 full adders are used in the adder tree and 153 full adders are used in the *CM* accumulation stage. For comparative evaluation it is useful to note that the total full adder count of the MF1BT [4] based ME hardware architecture presented in [17] is 153, as only a single bit-plane is used in the matching process (but the ME accuracy is lower). On the other hand the hardware architecture presented in [11] requires 6368 full adders to compute the SAD of a macroblock. Therefore, the hardware complexity of the proposed approach is in between 1BT based architectures and 8 bits/pixel representation based architectures, but close to 1BT based architectures.

Synthesis of the proposed architecture is performed using the Synplicity Synplify Pro synthesis tool with all advanced features such as retiming and pipelining turned off, so as to provide comparison with the architecture presented in [17] for the ME approach proposed in [4]. The total number of CLBs (Configurable Logic Blocks) occupied by the proposed architecture on a Xilinx XC2VP30 device is 2339, while this number is 690 for the architecture presented in [17]. Therefore, the hardware size is increased roughly 3.4 times whereas the depth of image pixels used in the matching process is increased 3 times. Thus, the hardware complexity seems increase about linearly with the pixel bit-depth.

IV. Experimental Results

In the experimental setup, initially, the motion estimation performance of the proposed truncated gray-coded bit-plane matching based ME approach (T-GCBPM) is evaluated using an open loop scheme, in which the current frame of the video sequence is reconstructed from the previous frame using motion vectors obtained by the ME approach. The similarity between the original and the estimated frames are computed in terms of Peak Signal to Noise Ratio (PSNR). Six different video sequences are utilized in the experiments to properly assess the performance of the proposed approach.

Average PSNR values for the test sequences are given in Table II. Here, T-BPM represents the conventional truncated bit-plane matching approach presented in [11]. In case of T-BPM and T-GCBPM, experimental results are provided for various *NTB* cases.

Experimental results show that the proposed T-GCBPM based ME approach outperforms conventional T-BPM based ME in all *NTB* cases. The increase in PSNR can be as high as 0.5dB. These results show that Gray-coding improves the performance of truncated bit-plane matching. Experimental results also show that the proposed T-GCBPM based ME approach with *NTB*=5 provides higher PSNR values compared to other low bit-depth ME approaches such as C-1BT or 2BT.

TABLE II. Average PSNR Values (dB) For Several Test Sequences Using an Open-Loop Scheme

	Video Sequences					
Method	Football (352x240) (125 frames)	Flowergarden (352x240) (115 frames)	Mobile (352x240) (140 frames)	Tennis (352x240) (112 frames)	Coastguard (352x288) (300 frames)	Foreman (352x288) (300 frames)
SAD, 8 bits/pixel	22.88	23.79	22.99	29.87	30.48	32.11
1BT [3]	21.83	23.32	22.71	28.77	29.84	30.44
2BT [5]	22.08	23.43	22.72	28.89	29.93	30.71
MF-1BT [4]	21.81	23.26	22.73	28.78	29.88	30.38
C-1BT [6]	22.10	23.39	22.77	29.18	29.98	30.87
T-BPM [11] (*NTB*=6)	22.08	23.49	22.72	28.57	28.97	30.35
T-BPM [11] (*NTB*=5)	22.22	23.49	22.75	28.68	29.85	30.86
T-BPM [11] (*NTB*=4)	22.22	23.49	22.76	28.70	29.95	31.04
T-BPM [11] (*NTB*=3)	22.21	23.48	22.76	28.70	29.95	31.08
T-BPM [11] (*NTB*=2)	22.20	23.48	22.76	28.70	29.95	31.09
T-GCBPM (*NTB*=6)	22.39	23.61	22.84	28.98	29.14	30.64
T-GCBPM (*NTB*=5)	**22.59**	**23.67**	**22.86**	**29.19**	**30.16**	**31.32**
T-GCBPM (*NTB*=4)	22.58	23.66	22.87	29.26	30.27	31.57
T-GCBPM (*NTB*=3)	22.56	23.66	22.87	29.23	30.26	31.61
T-GCBPM (*NTB*=2)	22.56	23.66	22.87	29.23	30.25	31.61

This increase in PSNR can be accounted to the fact that three bit-planes are used in the matching process for T-GCBPM based ME with *NTB*=5, while only two bit-planes are used in the matching process in case of C-1BT and 2BT based ME. The proposed T-GCBPM also has a lower binarization complexity compared to 1BT and 2BT based approaches.

The proposed hardware is coded in Verilog hardware description language and verified for a clock frequency of 90 MHz using synthesis with the Synplicity Synplify Pro synthesis tool.

The synthesized design occupied 2339LUTs (8%) on a Xilinx XC2VP30 device. The power consumption of the proposed T-GCBPM based ME hardware architecture is obtained to be about 230 mW on average, while the power consumption for T-BPM based ME is obtained as about 245 mW on average, for the mobile sequence at a clock frequency of 66MHz. Therefore, the proposed T-GCBPM based ME architecture results in a small reduction in power consumption compared to T-BPM based ME, in addition to improved ME accuracy. Note that the XPower and ISE Simulator (ISIM) tools from Xilinx are used for the power consumption analysis. The proposed hardware architecture is also synthesized for a 0.18um process to compare the performance with available 8bits/pixel based ME hardware architectures. According to the synthesis results, the gate count for the 1-bit/pixel based hardware architecture proposed in [17] is 8k, the number of total gates for the hardware architecture proposed in this work is 23k gates. In [11] where the least significant 3 bits of pixels are truncated to reduce ME hardware complexity, the total number of gates for fixed block size ME is 88k. In [26], the total gate count utilized for integer motion estimation is 146k gates, which is roughly 6 times the size of the proposed hardware architecture. Compared to 1-bit/pixel based architectures, the proposed hardware architecture requires about three times more gates, which is directly the result of using 3 bits/pixel instead of 1 bit/pixel representations, but the proposed hardware architecture has two important advantages: first of all the ME accuracy of the proposed approach is much better (the PSNR of reconstructed frames is up by nearly 1 dB) and secondly the proposed approach only requires Gray coding of pixel values which is a very low-complexity process and can directly be applied on a pixel-by-pixel basis, whereas 1BT and 2BT based approaches require a comparatively more complex transform process which introduces substantial additional hardware complexity that is not accounted for in the presented results. Compared to recently proposed 8bits/pixel based ME architectures utilizing SAD based matching criterion, the hardware complexity of the proposed approach is dramatically lower, making the proposed approach particularly favorable for consumer electronics applications that require low complexity and low power consumption.

V. Conclusions

A novel gray coded bit plane matching based ME approach with bit truncation is proposed in this paper. The proposed truncated gray-coded bit-plane matching based ME approach is shown to provide improved motion estimation accuracy compared to conventional bit-plane matching based ME. Furthermore, the presented approach also outperforms low bit-depth representation based ME methods, such as 1BT and 2BT based ME, in terms of ME accuracy; and also has a lower binarization complexity. The binarization complexity is much lower because the proposed approach uses simple Gray-coding that has very low-complexity and can be applied on a pixel-by-pixel basis, whereas 1BT and 2BT based approaches require much more complex transformation processes that actually add to the hardware complexity. An efficient hardware architecture of the proposed method is designed and verified in this paper. The proposed approach is particularly suitable for consumer electronics equipment with low processing resources and limited power capabilities.

References

[1] ITU-T Recommendation H.263, “Video coding for low bit rate communication,” 1996.

[2] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, “Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC),” *JVT-G050*, March, 2003.

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

[4] 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.

[5] A. Ertürk and S. Ertürk, “Two-Bit Transform for Binary Block Motion Estimation,” *IEEE Trans. Circuit Syst. Video Technol.*, vol. 15, no. 7, pp. 938- 946, July 2005.

[6] O. Urhan and S. Ertürk, “Constrained one-bit transform for low-complexity block motion estimation,” *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 17, no.4, pp. 478-482, April 2007.

[7] O. Urhan, “Constrained one-bit transform based motion estimation using predictive hexagonal pattern,” *Journal of Electron. Imaging*, vol. 61, no. 3, Article ID: 033019, July-Sep. 2007.

[8] H. Lee, J. Jeong, “Early termination scheme for binary block motion estimation,” *IEEE Trans. Consumer Electron.*, vol. 53, no. 4, pp. 1682-1686, Nov. 2007.

[9] S. Lee, J.M. Kim, S.I. Chae, “New motion estimation algorithm using adaptively quantized low bit-resolution image and its VLSI architecture for MPEG2 video encoding,” *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 8, no. 6, pp. 734-744, Oct. 1998.

[10] Y.W. Huang, T.C. Wang, B.Y. Hsieh, L.G. Chen, “Hardware Architecture Design for Variable Block Size Motion Estimation in MPEG-4 AVC/JVT/ITU-T H.264,” Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), vol. 2, pp. 796-799, May 2003.

[11] C.Y. Chen, W.Y. Chien, Y.W. Huang, T.C. Chen, T.C. Wang, L.G. Chen, “Analysis and Architecture Design of Variable Block-Size Motion Estimation for H.264/AVC,” *IEEE Tran. Circuits Syst.* vol. 53, no. 2, pp.578-593, Mar. 2006.

[12] T.-C. Chen, Y.-H. Chen, S.-F. Tsai, S.-Y. Chien, L.-G. Chen, “Fast Algorithm and Architecture Design of Low-Power Integer Motion Estimation for H.264/AVC,” *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 17, no. 5, pp. 568-577, May 2007.

[13] Z. Chen, T. Ikenaga, S. Goto, “A Hardware/Software Co-solution to Achieving High Throughput Required by Motion Estimation Part in H.264/AVC HDTV Real-time Application,” Proc. of IEEE International Symposium on VLSI Design, Automation and Test, VLSI-DAT, pp. 128-131, Apr. 2008.

[14] 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.

[15] S.-H. Wang, W.-L. Tao, C.-N. Wang, W.-H. Peng, T. Chiang, “Platform based design of all binary motion estimation (ABME) with bus interleaved architecture,” Proc. of IEEE International Symposium on VLSI Design, Automation and Test, pp. 241-244, Apr. 2005.

[16] E.A. Al Qaralleh, T.-S. Chang, K.-B. Lee, "An Efficient Binary Motion Estimation Algorithm and its Architecture for MPEG-4 Shape Encoding," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 16, no. 7, pp. 859-868, Jul. 2006.

[17] A. Çelebi, O. Urhan, I. Hamzaoğlu, S. Ertürk, "Efficient Hardware Implementations of Low Bit Depth Motion Estimation Algorithms," *IEEE Signal Process. Letts.*, vol. 16, no. 6, pp. 513-516, June 2009.

[18] A. Çelebi, O. Akbulut, O. Urhan, I. Hamzaoğlu, S. Ertürk, "An All Binary Sub-Pixel Motion Estimation Approach and its Hardware Architecture," *IEEE Trans. Consumer Electron.*, vol. 54, no. 4, Nov. 2008.

[19] Y. Baek, H.S. Oh, H.K. Lee, "An efficient block-matching criterion for motion estimation and its VLSI implementation," *IEEE Trans. Consumer Electron.*, vol. 42, no. 4, pp. 885-892, Nov. 1996.

[20] A. Bahari, T. Arslan, A.T. Erdogan, "Low power variable block size motion estimation using pixel truncation," Proc. of. IEEE International Symposium on Circuits and Systems, pp. 3663-3666, New Orleans USA, May 2007.

[21] K.M. Yang, M.T. Sun, , L. Wu, "A Family of VLSI Designs for the Motion Compensation Block-matching Algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, no. 2, 1317–1325, 1989.

[22] Z.-L. He, C.-Y. Tsui, K.-K. Chan, M. L. Liou, "Low Power VLSI Design for Motion Estimation Using Adaptive Pixel Truncation," *IEEE Trans. Circuits and Syst. Video Technol.,* vol. 10, no. 5, pp. 669-678, Aug. 2000.

[23] S.J. Ko, S.H. Lee, S.W. Jeon, E.S. Kang, "Fast digital image stabilizer based on Gray-coded bit-plane matching," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 598-603, Aug. 1999.

[24] O. Urhan, and S. Ertürk, "Gray-coded bit-plane matching for block based motion estimation," Proc. of 10th Signal Processing and Communication Applications Conference, vol. 1, pp. 518-523, Denizli, Turkey, June 2002. (In Turkish)

[25] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, 2nd., Kluwer Academic Publishers, London, 1997

[26] Y.-H. Chen, C.-C. Cheng, T.-Z. Chuang, C.-Y. Chen, and L.-C. Chen, "Efficient Architecture Design of Motion-Compensated Temporal Filtering/Motion Compensated Prediction Engine," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 18, no. 1, pp. 98-109, Jan 2008.

Anıl Çelebi (S'00) was born in Ordu, Turkey. He received the B.Sc., M.Sc. and Ph.D. degrees in electronics and communication engineering from Kocaeli University, Kocaeli, Turkey, in 2002, 2005, and 2008, respectively. Since 2002 he has been with the Department of Electronics and Telecommunications Engineering, University of Kocaeli, Turkey, where he is currently Assistant Professor. His research interests include very large scale integration (VLSI) design and implementation for analog/mixed signal systems, image processing systems, and video coding systems.

Orhan Akbulut was born in Kütahya, Turkey. He received the B.Sc. and M.Sc. degrees in electronics and telecommunication engineering from Kocaeli University in 2005 and 2007 respectively. He is currently working towards the Ph.D. degree at the Graduate School of Natural and Applied Sciences, Kocaeli University. His major research interests are image and video coding systems.

Oğuzhan Urhan (S'02-M'06) received his B.Sc., M.Sc., and Ph.D. degrees in electronics and telecommunication engineering from the University of Kocaeli, Kocaeli, Turkey, in 2001, 2003, and 2006, respectively.

Since 2001 he has been with the Department of Electronics and Telecommunications Engineering, University of Kocaeli, Turkey, where he is currently Associate Professor. He was a Visiting Professor at Chung-Ang University, Korea, from 2006 to 2007. His research interests include digital signal and image processing, in particular, image and video restoration and coding.

Sarp Ertürk (M'99) received his B.Sc. in Electrical and Electronics Engineering from Middle East Technical University, Ankara in 1995. He received his M.Sc. in Telecommunication and Information Systems and Ph.D. in Electronic Systems Engineering in 1996 and 1999 respectively from the University of Essex, U.K. From 1999 to 2001 he carried out his compulsory service at the Army Academy, Ankara. He is currently appointed as Full Professor at Kocaeli University, where he worked as Assistant Professor between 2001 and 2002, and Associate Professor between 2002 and 2007. His research interests are in the area of digital signal and image processing, video coding, remote sensing and digital communications.