Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 2206 – 2217
© TÜBİTAK
doi:10.3906/elk-1711-399

Adaptive bit-plane selection-based low complexity motion estimation for screen content coding

Ramazan DUVAR, Oğuzhan URHAN*
Department of Electronics and Telecommunication Engineering, Faculty of Engineering, Kocaeli University, Kocaeli, Turkey

Received: 28.11.2017 • **Accepted/Published Online:** 06.07.2018 • **Final Version:** 28.09.2018

Abstract: Screen content video coding has become an emerging research topic with the spread of applications such as cloud gaming, screen/desktop virtualization, and mobile or external display interfacing. Screen content videos have different features compared to conventional camcorder-captured scenes. In this work, a novel low bit-depth representation-based motion estimation approach is proposed to exploit screen content specific features to improve coding efficiency. The proposed approach is based on an adaptive selection of Gray-coded bit-planes in order to generate low bit-depth representation of original screen content frames. The experimental results show that the motion estimation performance of the proposed approach is significantly better compared to the methods in the same category for screen content videos.

Key words: Screen content coding, low-bit depth representation, motion estimation

1. Introduction

Today, screen content coding (SCC) has become an inevitable part of video coding applications. SCC is accepted as an extension for High Efficiency Video Coding (HEVC) [1]. Applications such as cloud gaming, screen/desktop visualization, and wireless display are examples of screen content. There are many differences between conventional camcorder-captured videos and screen content. For example, screen content is usually noiseless and includes sharp edges [2]. These characteristic features of screen content might be used to improve the video coding efficiency. The computational complexity of the algorithms used for SCC should be low for real-time applications like screen/desktop visualization.

The prediction mechanism is the core part of video compression approaches currently used. Intra- and interprediction techniques are used to benefit from spatial and temporal redundancy available in the video, respectively. There are different approaches in the literature for screen content coding to exploit these redundancies.

In [2], a nontransform coding scheme based on separating screen content into color and structure components was proposed for screen content. A transform skipping mode for screen content coding in HEVC was proposed in [3] since the intracoding stage performs well for screen content. A color table and index map-based coding scheme was presented in [4], which improves intraprediction accuracy by up to 26%.

A fast intrablock copy search approach was proposed in [5] as a block matching technique for intracoding. Block matching was used to remove redundancy from repeating patterns. In [6], an intraprediction method was

*Correspondence: urhano@kocaeli.edu.tr

proposed where intraprediction directions are selected according to edge positions and directions. A hash-based block matching method for screen content was proposed in [7] where both intra- and interprediction approaches are employed.

The motion estimation (ME) process for interprediction is performed by searching the nonoverlapped blocks in a search window defined in the reference frame(s). In full search (FS)-based ME, each block in the current frame is searched by controlling all possible candidate locations within the search range. In order to measure block similarity, it is possible to utilize the sum of squared differences (SSD) criterion between the original and candidate block as in Eq. (1).

$$SSD\left(m,n\right)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left\{I^{t}\left(i,j\right)-I^{t-1}\left(i+m,j+n\right)\right\}^{2},\,-s\leq mn\leq s \tag{1}$$

Here, I^t and I^{t-1} denote the current and reference frames. In this equation $(m,n)\,,s\,,$ and N represent the candidate motion vector location, search range, and block size, respectively. It might be possible to employ the sum of absolute differences (SAD) criterion to eliminate the computational burden of square operation. Even though these matching criteria provide the best motion estimation accuracy when used in combination with the FS scheme, they are not suitable for real-time applications because of their computational complexity.

In the literature, there are many studies aiming to reduce the computational complexity of the ME process. Three-step search [8], diamond search [9], and hexagonal search [10] approaches aim to alleviate the computational complexity of FS by making use of some specific locations instead of all candidates as in the FS approach. An advanced form of these approaches called the test zone search [11] is adopted in the HEVC reference software for fast ME.

Another group of ME approaches aim to reduce the computational complexity of matching criterion computation employing a low bit-depth version of the original image frames for the block matching process. In this case, matching computation can be carried out using simple Boolean operations efficiently compared to the SSD computation in Eq. (1). Since the bit-depth of the input image is reduced, it might be possible to improve parallelism to further speed up ME computations. These kinds of approaches initially convert full bit-depth images to low bit-depth using a filtering or selection mechanism. Next, the motion estimation is carried out using these low bit-depth resolution images at all candidate locations using the number of nonmatching points (NNMP) criterion. The effectiveness of hardware implementation of these approaches is shown in the literature compared to conventional SAD or SSD.

One-bit transform-based block motion estimation [12] (1BT), multiplication-free one-bit transform-based block motion estimation [13] (MF-1BT), constrained one-bit transform-based block motion estimation (C-1BT) [14], truncated Gray-coded bit-plane matching (T-GCBPM) [15], selective Gray-coded bit-plane matching-based motion estimation (SGCME) [16], and fast binary motion estimation for screen content (FBMESC) [17] are examples of this kind of approaches. To our knowledge, there is only one approach (i.e. [17]) that aims to employ a low bit-depth representation-based method for SCC.

In this work, we present a novel low bit-depth representation-based ME method specifically designed for screen content. The proposed approach employs Gray-coded bit-planes according to the level of details in the planes and constructs only 1-bit depth images for matching computation. At the matching stage, bit-planes are weighted according to their significance to improve the ME accuracy.

In the next section, conventional low-bit depth representation-based approaches are introduced. The details of the proposed approach are presented in Section 3. In Section 4, experimental results are provided, and conclusions are given in Section 5.

2. Low bit-depth-based motion estimation approaches

In low bit-depth representation-based ME approaches, initially the full bit-depth original image frames are converted to low bit-depth images using transforms such 1-BT [12] or MF-1BT [13]. For example, the 1BT-based ME approach presented in [12] initially applies a multiband-pass filter to input frames. The filter kernel is given in Eq. (2). Next, the binary image is constructed by comparing the original frame with the filtered frame as in Eq. (3).

$$K\left(i,j\right) = \begin{cases} \frac{1}{25} \leftarrow i,j \in [1,4,8,12,1] \\ 0 \leftarrow i,j \notin [1,4,8,12,16] \end{cases} \tag{2}$$

$$B(ij) \begin{cases} 1 \leftarrow I(i,j) \geq I_f(i,j) \\ 0 \leftarrow I(i,j) < I_f(i,j) \end{cases} \tag{3}$$

Here, I and I_f denote original and filtered frames, respectively. A sample image from the "Big Bunny" sequence with its filtered and binary versions are shown in Figures 1a, 1b, and 1c, respectively. After the binarization operation, the current frame is divided into nonoverlapping blocks and each block is searched within a search window in reference frame(s). Conventionally, NNMP is used as a matching criterion and is computed as follows.

(a) (b) (c)

Figure 1. a) Sample frame of the Big Bunny Sequence, b) filtered frame using the kernel in Eq. (3), c) obtained frame using Eq. (4) (1BT).

$$NNMP\left(m,n\right) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left\{ B^t\left(i,j\right) \oplus B^{t-1}\left(i+m,j+n\right) \right\} \tag{4}$$

Here, B and B^{t-1} are the low bit-depth version of the current and reference frame. In Eq. (4), $\oplus$ denotes Boolean EX-OR operation. This computation is performed at all candidate motion vectors positions in the search window. The position providing the smallest NNMP is decided as the motion vector of the current block [12–14].

One of the major drawbacks of 1BT and MF-1BT-based ME approaches is that pixels that have similar intensity values might be assigned to different categories during the binarization process in Eq. (3). This unwanted effect might reduce motion estimation accuracy. In order to alleviate this problem, a constraint mask was introduced in [14] where pixels close in value are discarded in the matching. For this purpose, a constraint mask (CM) is generated as in Eq. (5) by making use of simple thresholding.

$$CM(ij)\begin{cases} 1 \leftarrow |I(i,j) \geq I_f(i,j)| \geq T \\ 0 \leftarrow |I(i,j) < I_f(i,j)| < T \end{cases} \tag{5}$$

In Eq. (5), T denotes a fixed threshold value. The matching criterion in C-1BT is constructed as in Eq. (6) to exclude unreliable pixels from the computation by the help of CM.

$$CNNMP\,(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\begin{cases} \left[CM^t\,(i,j)\,||CM^{t-1}\,(i+m,j+n)\right] \\ \cdot\, B^t\,(i,j) \oplus B^{t-1}\,(i+m,j+n)\right] \end{cases} \tag{6}$$

Here, $||$ shows Boolean OR whereas $\cdot$ denotes Boolean AND operation.

In general, 1BT, MF-1BT, and C-1BT-based ME approaches require computationally complex filtering operation for binarization. Thus, investigation of computationally lightweight binarization approaches is important to fully benefit from the advantages of EX-OR-based matching criteria computations in the ME process. The following approaches essentially aim to obtain the binary representation of the original image frames at lower complexity.

In [15], the T-GCBPM approach was presented. After converting binary values to Gray-coded values via simple EX-OR based operations a truncation is performed to use a few of the most significant bits. The matching criterion for this approach is computed as follows.

$$MC\,(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{p=NTB}^{K-1} 2^{k-NTB} \times \left\{ GC_p^t\,(i,j) \oplus GC_p^{t-1}\,(i+m,j+n)\right\} \tag{7}$$

Here, GC is the Gray-coded frame, p is the bit-plane number, K is the total number of bit-planes, and NTB is the number of truncated bit-planes. For example, if NTB is set to 5, this means that only the most significant 3 bit-planes will be used in matching criterion computation. In [18], a single bit-plane was generated using four bit-planes by placing them in regular order and NNMP was utilized as the matching criterion for this method. This approach was designed for global motion estimation and is not suitable for local motion estimation.

In [16], a combination of the methods proposed in [15] and [18] was presented where the most significant three bit-planes are placed as in Figure 2 to construct the binary image. In this way, strong parts of these approaches are used together to improve ME accuracy and reduce computational complexity. It is important to note that during the NNMP computation in [16] and [18], the same selection approach is applied to all candidate locations independently.

Recently, a novel approach to applying low bit-depth representation-based ME methods to screen content videos was presented in [17]. This method aims to choose a single bit-plane for each reference block according to the level of details in that bit-plane using natural binary codes. For this purpose, initially, edge maps are generated for each bit-plane starting from higher bit-planes. If there is not enough detail in the current bit-plane, then the next bit-plane is investigated. The edge maps are constructed as follows.

$$M_p\,(i,j) = \begin{cases} 1 \leftarrow \exists\,(m,n) \in \{(\pm 1,0)\,,(\pm 1,0)\}\,, BC_p(i,j) \oplus BC_p(i+m,j+n) \\ 0 \leftarrow \;\text{otherwise} \end{cases} \tag{8}$$

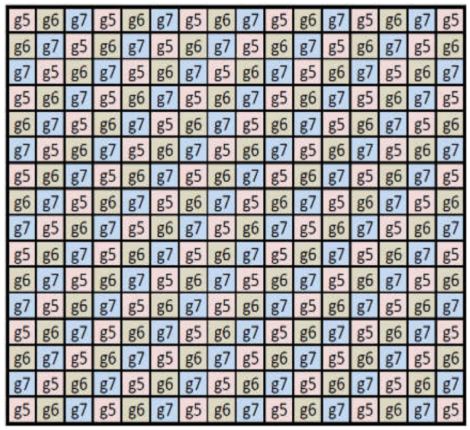
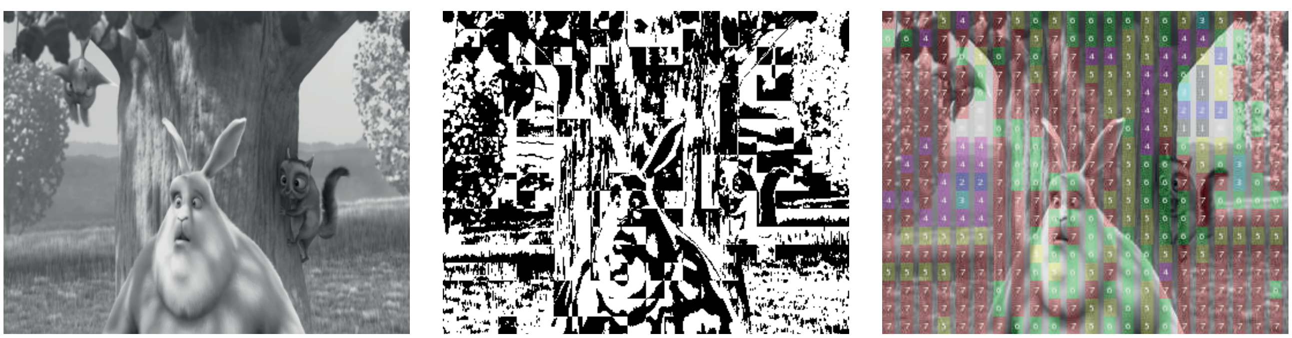


Figure 2. Bit-plane selection approach for a 16 $\times$ 16 block presented in [16].

Here, BC is the binarily coded frame, M is the edge map, and p is the bit-plane number. If the number of 1s in M is higher than a fixed threshold then this bit-plane is considered as reliable enough for matching computation. Otherwise, the same procedure is repeated for subsequent bit-planes until the number of 1s in that bit-plane is higher than the threshold. In Figure 3 a sample frame is provided from the "Big Bunny" sequence (Figure 3a) together with the constructed binary version of this frame (Figure 3b) and an illustration that shows which bit-plane is selected (Figure 3c) according to the edge map-based approach in [17]. As seen from this figure, lower bit-planes are chosen when the block is homogeneous, which in the end may reduce ME performance since these bit-planes generally contain noisy data.



(a) (b) (c)

Figure 3. a) Sample frame of the Big Bunny sequence, b) constructed frame with selected bit-planes of blocks using the method in [17], c) selected bit-planes for blocks.

3. Proposed binarization approach

The proposed approach introduces the following novelties: adaptive selection of Gray-coded bit-planes for matching according to the edge map generated from the 7th Gray-coded bit-plane and weighted utilization of each Gray-coded bit-plane in matching computation. These novelties not only improve ME accuracy compared to the methods in [16] and [17], but also significantly reduce the computation load of the method in [17].

The method in [16] combines one of the three most significant Gray-coded bit-planes as in Figure 2 for each pixel position to generate a single binary image for matching criterion computation. Next, NNMP computation is carried out to decide motion vectors. On the other hand, the binarization approach presented in [17] aims to improve ME accuracy of previous low bit-depth-based ME approaches by separately selecting a single binary coded bit-plane for each block. However, our experiments show that this block-based binarization approach is not able to decide the correct bit-plane in some circumstances. Thus, its performance may be degraded because of the incorrect selection of bit-planes.

In this work, we aim to benefit from strong parts of the methods presented in [16] and [17] and improve them in a novel way. The advantage of using Gray-coded bit-planes compared to natural binarily coded ones was discussed in [15,16,19]. In general, Gray codes have a single digit change in successive pixel intensity values and thus they are more suitable for matching computation. Hence, we prefer to utilize Gray-coded bit-planes in our method.

It is important to decide to the number of bit-planes that will be included in matching computation. The method in [16] employs only the most significant three bit-planes, as in Figure 2, whereas in [17] it is possible to employ one of the available bit-planes according to the edge map. Figures 4a–4h show all Gray-coded bit-planes for "Big Bug Bunny" and "Google Maps" sequences. As seen from Figure 4a and Figure 4e, the most significant bit-planes may not contain reliable information for matching by themselves, especially for the flat image regions where neighbor pixels have the same binary values and thus there is not any discriminating structure for block matching in higher bit-planes, whereas lower bit-planes may include some noise like binarization effect in certain regions, as in Figure 4d. However, as shown in [15], the most significant four bit-planes generally provide enough information for form matching.

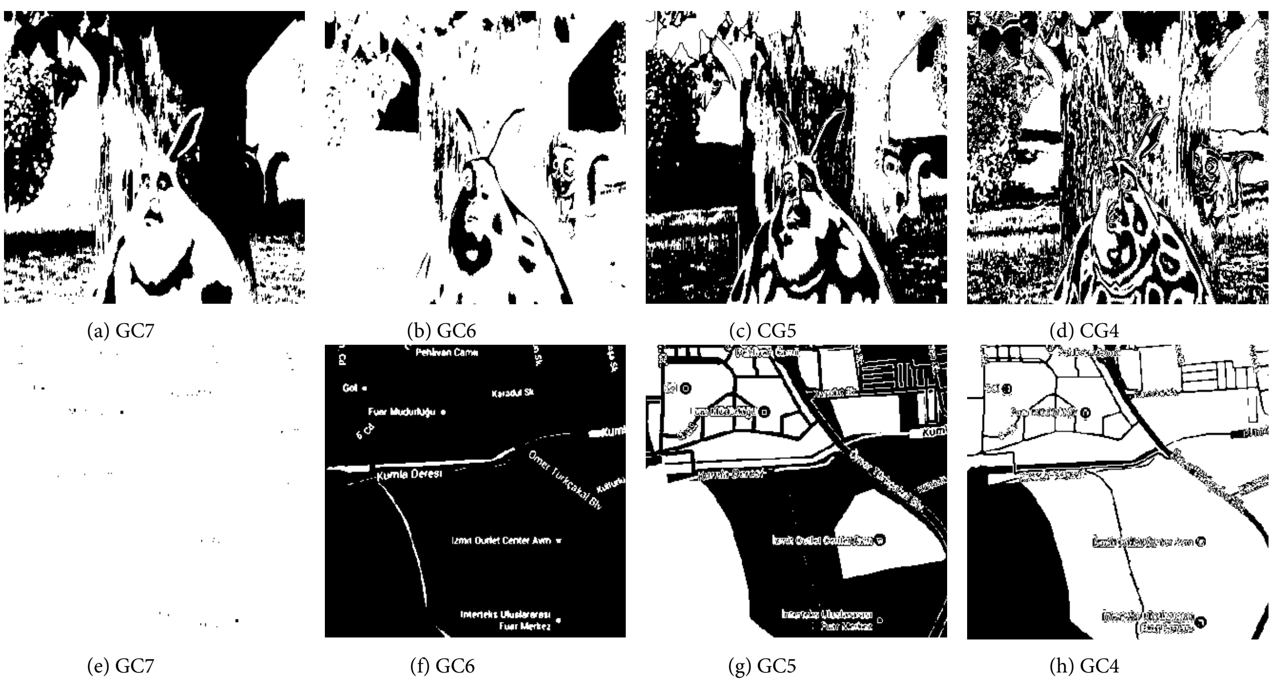


Figure 4. Gray-coded bit-planes of the sample frames of Big Bunny. GC7, GC6, GC5, GC4 of a) - d) "Big Bug Bunny" and e) - h) "Google Maps" sequences.

In this paper, we propose to select the Gray-coded bit-planes to be included in matching criterion computation by simply checking the edge map at the most significant bit-plane only. If the number of 1s in the edge map's 7th Gray-coded bit-plane is higher than a fixed threshold (called Case-A), we decide to utilize only the most significant three bit-planes since the amount of details would be enough for matching in this case. Otherwise, we include the 4th bit-plane in the generation of binary frames (Case-B) as shown in Figure 5 to include more details in the constructed 1-bit depth image.

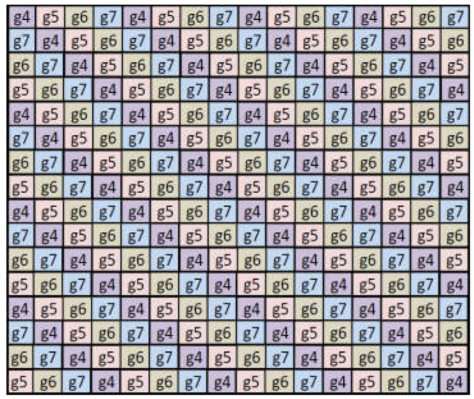


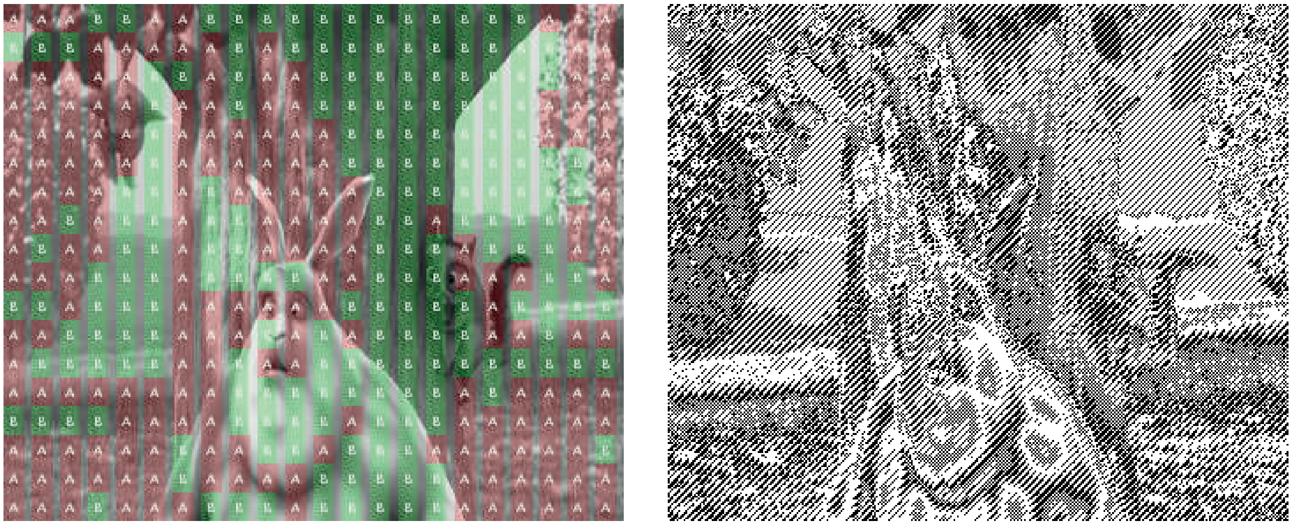
Figure 5. Bit-plane selection approach for a 16 $\times$ 16 block in the case of low amount of details in 7th Gray-coded bit-plane.

Figure 6a shows the distribution of Case-A and Case-B for a sample image frame from the "Big Bunny" sequences. As seen from Figure 6a, the proposed approach generally employs three bit-planes for textured regions, whereas four bit-planes are utilized for smooth areas as expected. The constructed binary image of Figure 3a by the proposed method is shown in Figure 6b. As seen from Figure 6b, the proposed approach is able to capture all useful details for matching. The proposed method significantly reduces the computational complexity originating from the edge map computations of the method in [17]. Note that the method in [17] continues the edge map computation until it finds enough details at the evaluated bit-plane, which results in higher computational complexity. Table 1 shows the utilization of bit-planes in the edge-map computation for five different image sequences. As seen from this table, the proposed approach is able to reduce the computational load at this stage by up to 82% and the average reduction in computational complexity is about 70%.

After the original image frame is converted into a binary image as in Figure 6b, the next step is to

Table 1. Number of edge map calculated blocks.

Selected plane	Seq. BB	Seq. ED	Seq. CS	Seq. PES	Seq. GM	Average
7	51,360	7134	33,255	82,837	4745	35,866
6	20,270	6388	22,004	10,929	7399	13,398
5	13,850	4222	18,420	13,398	25,930	15,164
4	8037	3110	4186	120,906	1919	27,632
3	1511	8215	1256	11,930	1346	4851
2	1583	10,561	63	0	1110	2663
1	985	14,649	13	0	292	3187
0	1404	44,771	3	0	36,459	16,527
Total	199,374	565,397	156,037	688,163	375,656	396,913
Proposed	99,000	99,000	79,200	240,000	79,200	119,288
Gain (%)	50.34	82.49	49.24	65.12	78.91	69.94



(a) (b)

Figure 6. a) Case selection for a sample frame from the "Big Bunny" sequence, b) constructed binary form of Figure 3a by the proposed method.

compute matching between the blocks in the current frame and reference frames. In this paper, we propose to utilize a novel matching criterion by weighting the Gray-coded bit-planes in single bit-depth binary images. The proposed measure is called the weighted selective Gray-coded bit-plane matching criterion (WSGC-MC) and is formulated as follows.

$$WSGC - MC\,(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} 2^{p-s} \times \left\{ SG_p^t\,(i,j) \oplus SG_p^{t-1}\,(i+m,j+n) \right\} \tag{9}$$

Here, SG is the constructed binary image frame by the proposed Gray-coded bit-plane selection mechanism and 2^{p-s} is the bit-plane-dependent weighting factor. Note that s is set to 4 or 5 according to our selection mechanism. Our experiments reveal that weighting higher bit-planes results in better ME accuracy.

4. Experimental results

The accuracy of motion estimation approaches is often measured in terms of the peak signal to noise ratio (PSNR) between the original and predicted frames. This approach is called as open-loop scheme and is generally adopted in the comparison of ME methods. We employ five different screen content-type image sequences in the experiments. These sequences are Big Bunny (BB - 250 frames) (352 $\times$ 288), Elephants Dream (ED - 250 frames) (352 $\times$ 288), Counter Strike (CS - 200 frames) (352 $\times$ 288), Pro Evolution Soccer 2013 (PES - 200 frames) (640 $\times$ 480), and Google Maps (GM - 200 frames) (352 $\times$ 288). Note that the CS, PES, and GM sequences are generated by us in this work and all these sequences are available on the supplementary website located at http://kule.kocaeli.edu.tr/ScreenContentSequences.

The Big Bunny sequence is generated from 3200th–3450th frames of the computer-generated "Big Buck Bunny" computer-animated comedy film, where only small to medium local motions exist. The Elephants Dream sequence is generated from the 6000th–6250th frames of the same-named computer-generated short film and contains big local motions and truck right camera movement. The Counter Strike sequence is captured from a video game and includes dolly-in movement and thus substantial global motion effect. The Pro Evolution Soccer 2013 sequence contains some significant camera movements together with the many small local motion effects caused by players in the field. The Google Maps sequence is generated from the Google's web map service. This sequence contains truck, boom, and significant amounts of zoom effects.

The PSNR results of the 1BT [12], C-1BT [14], T-GCBPM [15], SGCME [16], and FBMESC [17] methods together with the proposed method are given in Table 2. As seen from this table, the proposed method is not only able to outperform existing 1-bit depth representation-based methods such as 1BT [12], SGCME [16], and FBMESC [17] but also two-bit depth representation-based methods such as C-1BT [4] and T-GCBPM [15] (NTB = 6). It is also important to note that the proposed method provides similar ME accuracy compared to the NTB = 5 case where three separate bit-planes are used for matching at significantly higher complexity compared to the proposed method. NTB = 4 gives the best overall ME accuracy since it employs four different bit-planes in matching at the expense of higher computation load.

Table 2. PSNR performance (in dB) of different ME methods in open loop scheme.

Method	Seq. BB	Seq. ED	Seq. CS	Seq. PES	Seq. GM	Average
1BT [12]	34.63	29.00	24.37	29.83	30.29	29.62
C-1BT [14]	34.97	29.36	24.83	30.15	30.42	29.95
T-GCBPM [15] (NTB = 6)	35.59	30.78	25.10	29.81	27.50	29.76
T-GCBPM [15] (NTB = 5)	36.00	30.93	25.85	30.63	32.11	31.10
T-GCBPM [15] (NTB = 4)	36.00	30.90	25.91	31.33	32.60	31.35
SGCME [16]	35.69	30.56	25.55	30.47	31.47	30.75
FBMESC [17]	34.39	28.76	24.30	29.31	31.12	29.58
Proposed method	35.80	30.69	25.70	30.92	32.14	31.05

In order to evaluate the performance improvement achieved by the proposed method, we examine the effect of both different bit-plane selection schemes on the SGCME approach and weighting mechanisms. Table 3 shows the PSNR performance of SGCME-based methods for different configurations. As seen from this table, the inclusion of the 4th bit-plane in the SGCME-based method does not provide a constant performance gain for the different types of image sequences. For example, when the 4th bit-plane is taken into account, ME accuracy increases by about 0.3 dB for the Big Buck Bunny sequence, whereas it decreases by about 0.4 dB for the Google Maps sequence. Based on these observations, it is clear that the inclusion of the 4th bit-plane in the ME process has to be adaptive for the best possible ME performance. It is also clear from Table 3 that the weighting mechanism improves the ME performance of both bit selection approaches. However, similar to our previous observations, inclusion of the 4th bit-plane in the ME procedure for the weighted approach has varying effects on different sequences. For example, the performance of the WSGCME (5-6-7) configuration is better for Elephants Dream, whereas the WSGCME (4-5-6-7) configuration has better accuracy for Google Maps. When we evaluate the effectiveness of the proposed adaptive bit-plane selection mechanism according to the PSNR results in Table 3, it is clear that the proposed method is able to choose the appropriate bit-plane combination for matching.

We also present the computational complexity of the compared ME methods in Table 4 with their optimized software implementations. Note that the computational complexity of the FBMESC in [17] and the proposed method is content-dependent. As seen from this table, the proposed method has significantly lower

Table 3. PSNR performance (in dB) of different ME methods in open loop scheme.

Method	Seq. BB	Seq. ED	Seq. CS	Seq. PES	Seq. GM	Average
SGCME (5-6-7) [16]	35.69	30.56	25.55	30.47	31.47	30.75
SGCME (4-5-6-7)	35.40	30.30	25.38	30.91	31.86	30.77
WSGCME (5-6-7)	35.74	30.72	25.65	30.39	31.61	30.82
WSGCME (4-5-6-7)	35.76	30.59	25.66	30.93	32.11	31.01
Proposed method	35.80	30.69	25.70	30.92	32.14	31.05

transform stage (binarization) complexity compared to 1BT [12], C-1BT [14], and FBMESC [17] approaches. In the matching stage, the proposed method has comparable computational complexity with the ME methods with C-1BT [14] and SGCME [16] and significantly lower complexity compared to T-GCBPM [15].

Table 4. Average computational complexity of different ME methods (per pixel).

Method	Transform							Matching				
	Add.	Div.	Shift	Sub.	Comp.	Boolean Op.	Memory access	Boolean Op.	Shift	Add.	Comp.	Memory access
1BT [12]	25	1	-	-	1	-	28	1	-	1	-	2
C-1BT [14]	16	-	1	1	3	-	22	3	-	1	-	4
T-GCBPM [15] (NTB = 5)	-	-	3	-	-	3.666	6	3	2	3	-	6
SGCME [16]	1	-	1		2.666	1.666	3	1	-	1	2.666	2
FBMESC [17]	3.327	-	3.327	-	0.013	26.62	23.30	1	-	1	-	2
Proposed method	1	-	1	-	3.118	1.723	3.062	1	0.725	1	3.117	2

In general, when the ME performance of the proposed method given in Table 2 is assessed together with the computational complexity given in Table 4, the efficiency of the proposed method becomes clear. Only the T-GCBPM-based ME method in [15] with NTB = 5 and NTB = 4 has better PSNR performance compared to the proposed method. However, the computational load is significantly higher in the matching stage.

In order to evaluate ME performance against the computational time of the ME methods, we implement all ME methods in C++ at a similar optimization level. Next, we compute the total computational time taking only the transform and matching times into consideration. Figure 7 illustrates average frame PSNR for five test sequences used versus computational time required to perform transform and the matching part of motion vector computation for a 16 $\times$ 16 block on a PC with Core i7 2.2 GHz processor and 16 GB of RAM. In this graphic, the best method is expected to be located in the upper left corner, providing maximum ME accuracy in the lowest computation time. As seen from Figure 7, even though the 1BT-based [12] ME approach has the smallest computation time, its ME performance is the worst. On the other hand, T-GCBPM in [15] has the best ME performance while having the worst computation time. The proposed method has a good balance between ME accuracy and computation time, as seen in Figure 7.

Additionally, it is important to note that memory access requirements of the methods in [12,14,17] are at least seven times higher compared to the proposed approach, which might be a vital disadvantage when hardware implementation of these methods is considered.

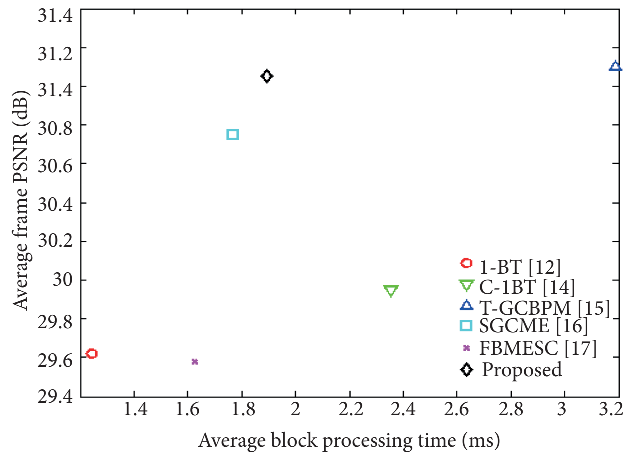


Figure 7. ME performance versus computational time.

5. Conclusions

In this work, a novel low bit-depth representation-based ME approach is presented for screen content coding. An edge map is generated from the most significant Gray-coded bit-plane, which is used to decide the Gray-coded bit-planes to be included in the matching criterion computation. Additionally, the proposed weighted selective Gray-coded bit-plane matching criterion enables better ME accuracy compared to the existing methods in the literature in terms of PSNR by providing lower computational complexity. It is important to note that the proposed approach can be implemented efficiently in single instruction multiple data (SIMD) infrastructures and possible hardware implementations and thus might enable real-time processing.

References

[1] Sullivan GJ, Ohm JR, Han WJ, Wiegand T. Overview of the high efficiency video coding (HEVC) standard. IEEE T Circ Syst Vid 2012; 22: 1648-1667.

[2] Zhu W, Ding W, Xu J, Shi Y, Yin B. Screen content coding based on HEVC framework. IEEE T Circ Syst Vid 2014; 16: 1316-1326.

[3] Mrak M, Xu JZ. Improving screen content coding in HEVC by transform skipping. In: 2012 European Signal Processing Conference; 27–31 August 2012; Bucharest, Romania. pp. 1209-1213.

[4] Ma Z, Wang W, Xu M, Yu H. Advanced screen content coding using color table and index Map. IEEE T Image Process 2014; 23: 4399-4412.

[5] Kwon DK, Budagavi M. Fast intra block copy (IntraBC) search for HEVC screen content coding. In: IEEE 2014 International Symposium on Circuits and Systems; 1–5 June 2014; Melbourne, Australia. pp. 9-12.

[6] Hu S, Cohen RA, Vetro A, Kuo JCC. Screen content coding for HEVC using edge modes. In: IEEE 2013 International Conference on Acoustics, Speech and Signal Processing; 26–31 May 2013; Vancouver, Canada. pp. 1714-1718.

[7] Zhu W, Ding W, Xu J, Shi Y, Yin B. Hash-based block matching for screen content coding. IEEE T Multimedia 2015; 17: 935-944.

[8] Koga T, Iinuma K, Hirano A. Motion compensated interframe coding for video conferencing. In: 1981 National Telecommunication Conference; 29 November–3 December 1981; New Orleans, LA, USA. pp. G5.3.1-G5.3.5.

[9] Zhu S, Ma KK. A new diamond search algorithm for fast block-matching motion estimation. IEEE T Image Process 2000; 9: 287-290.

[10] Zhu C, Lin X, Chau LP. Hexagon-based search pattern for fast block motion estimation. IEEE T Circ Syst Vid 2002; 12: 349-355.

[11] Parmar N, Sunwoo MH. Enhanced test zone search motion estimation algorithm for HEVC. In: 2014 International SoC Design Conference; 3–6 November 2014; Jeju, Korea. pp. 260-261.

[12] Natarajan B, Bhaskaran V, Konstantinides K. Low-complexity block-based motion estimation via one-bit transforms. IEEE T Circ Syst Vid 1997; 7: 702-706.

[13] Ertürk S. Multiplication-free one-bit transform for low-complexity block-based motion estimation. IEEE Signal Process Lett 2007; 7: 109-112.

[14] Urhan O, Ertürk S. Constrained one-bit transform for low-complexity block motion estimation. IEEE T Circ Syst Vid 2007; 17: 478-482.

[15] Çelebi A, Akbulut O, Urhan O, Ertürk S. Truncated gray-coded bit-plane matching based motion estimation and its hardware architecture. IEEE T Consum Electr 2009; 55: 1530-1536.

[16] Yavuz S, Çelebi A, Aslam M, Urhan O. Selective gray-coded bit plane based low complexity motion estimation and its hardware architecture. IEEE T Consum Electr 2016; 62: 76-84.

[17] Sun T, Wan P, Au OC, Dai W, Jia L, Yuan Y, Zheng A, Ma R. Fast binary motion estimation for screen content video coding. In: 2014 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference; 9–12 December 2014; Wat, Cambodia. pp. 1-5.

[18] Kuo TY, Wang CH. Fast local motion estimation and robust global motion decision for digital image stabilization. In: Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing; 15-17 August 2008; Harbin, China. pp. 442-445.

[19] Urhan O, Ertürk S. Gray coded bit-plane matching for block-based motion estimation. In: Proceedings of the 10th Signal Processing and Communication Applications Conference; 12–14 June 2002; Denizli, Turkey. pp. 518-523.