Constrained one-bit transform-based motion estimation using predictive hexagonal pattern

Oğuzhan Urhan
University of Kocaeli
Department of Electronics & Telecommunications Engineering
Kocaeli University Laboratory of Image and Signal Processing (KULIS)
Veziroglu Campus
41040, Kocaeli, Turkey
E-mail: urhano@ieee.org

Abstract. *We combine predictive hexagonal pattern and partial distortion searches with the recently proposed constrained one-bit transform-based motion estimation scheme to reduce the computational load of the motion estimation process. Furthermore, the kernel used to obtain the one-bit images is simplified. Experimental results show significant reduction of the number of average search points, with only a slight loss in motion estimation accuracy.* [DOI: 10.1117/1.2769321]

1 Introduction

Block matching-based motion estimation is a crucial part of many video coding schemes as it exploits temporal correlation and therefore reduces temporal redundancy between consecutive image frames. Full search (FS) is regarded as the optimal search procedure, but it suffers from extensive computational load. Many approaches have been proposed in order to reduce the computational burden of full search so as to enable fast computation of motion vectors introducing minimum level of error.

Many schemes proposed to reduce the computational load of full-search motion estimation use a predefined search pattern to reduce the number of search points. Among these, the diamond search- and hexagonal search-based approaches proposed in Refs. 1 and 2, respectively, provide acceptable results using fewer search points compared to other well-known sparse search point approaches. The method presented in Ref. 3 aims to speed up the diamond search-based approach, making use of motion vector and matching error information from neighboring blocks. The global motion of three previous frames is used in combination with motion vectors of neighboring blocks of the current block to reduce the computational load in Ref. 4. In Ref. 5, an enhanced version of the method presented in Ref. 2 is proposed by making use of the spatial correlation between neighboring blocks (referred to as predictive hexagonal search) and also a clever inner search procedure. It is shown that improved motion estimation results can be achieved using these modifications.

Lower bit-depth representations such as one-bit transform (1BT),[6] two-bit transform (2BT),[7] multiplication-free one-bit transform (MF-1BT),[8] and constrained one-bit transform (C-1BT)[9] are proposed to enable faster computation of the matching criteria using Boolean exclusive-OR (XOR) operations. These approaches can be executed faster than conventional sum of absolute differences (SAD) and minimum squared error (MSE) matching criteria, especially using hardware implementations, at the expense of a slight drop in accuracy. It is stated in Ref. 6 that the one-bit transform-based motion estimation provides roughly 15 times' improvement in speed. The modified approaches proposed in Refs. 10 and 11 aim to improve the motion estimation accuracy of 1BT- and 2BT-based motion estimation methods, by refining the motion vectors found by 1BT and 2BT using additional local searches in the image domain. However, these approaches require the computation of an image domain matching criteria (such as SAD) that prevents an all-binary implementation. The successive elimination algorithm (SEA) is combined with 1BT-based motion estimation in Ref. 12 to reduce the computational load of 1BT-based motion estimation. It is shown that the computational load can be reduced by up to 50% as a result of the SAE combination without any performance loss.

Despite intensive research in this area, the combination of a sparse search point procedure with a lower-bit representation approach has not been investigated in the literature so far, except possibly the plain unification of 1BT with logarithmic search as presented in Ref. 6, which, however, gives a considerably lower performance. The main novelty of this work is to combine a sparse search point approach with a lower-bit representation motion estimation method and show that this can be accomplished without significant performance loss.

2 1BT and C-1BT-Based Motion Estimation

In the 1BT-based motion estimation approach proposed in Ref. 6, image frames are initially filtered with a multiband-pass filter kernel, the result of which is compared against the original image frame to obtain binary images that are used in the motion estimation matching process. If I is used to denote the original image frame, and I_F represents the filtered image, the 1BT is constructed as

Paper 07012RR received Jan. 22, 2007; revised manuscript received Mar. 28, 2007; accepted for publication Mar. 30, 2007; published online Aug. 6, 2007.
1017-9909/2007/16(3)/033019/5/$25.00

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

The block-matching measure used in 1BT-based motion estimation is the number of non-matching points (NNMP) that basically use XOR operations for matching. Therefore, it can be computed much faster than the SAD criteria conventionally used for image domain matching.

The 1BT-based motion estimation uses a kernel size of 17×17 elements with 25 nonzero components. The filtering operation is carried out by means of convolution of the filter kernel with input image frames. The kernel used in the 1BT approach can be formulated as follows:

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

It is clear from (2) that floating-point operations are required for normalization during filtering. On the other hand, the kernel proposed in the MF-1BT approach of Ref. 8, which is also used in the C-1BT-based approach,[9] has 16 nonzero components; therefore, the normalization can be performed using an integer shift operation. This kernel leads to a faster processing of image frames as it both reduces the number of addition operations and also enables the use of integer shift normalization.

Another difference of C-1BT compared to 1BT-based motion estimation is the introduction of a constraining mask (CM). The CM can be expressed in the form of

$$CM^t(i,j) = \begin{cases} 1, & \text{if } |I^t(i,j) - I_F^t(i,j)| \geq D, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

and is used to determine pixels that have close intensity values but are classified into opposite sides in the binary one-bit image. The CM value of a pixel will be "1" if the pixel value is at least a certain distance (D) away from the transform threshold (note that the transform threshold is actually defined by I_F). If a pixel has a "0" CM value, it can be considered as unreliable and is to be discarded in the search process to improve the accuracy. The constrained number of non-matching points (CNNMP) measure is defined as in (4) to include the CM into the computation of the error criteria.[9]

$$\text{CNNMP}(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left\{ \begin{array}{l} [CM^t(i,j) \parallel CM^{t-1}(i+m,j+n)] \\ \odot [B^t(i,j) \oplus B^{t-1}(i+m,j+n)] \end{array} \right\}, \quad -s \leq m,n < s. \tag{4}$$

Note that $\parallel$, $\odot$, $\oplus$, and s show Boolean OR, AND, XOR operations and the search range in (4), respectively.

3 The Proposed Method

The method presented in this paper mainly proposes the combination of the C-1BT-based motion estimation approach with a predictive hexagonal search procedure to further decrease the computational load of C-1BT-based motion estimation. Furthermore, it also proposes using a new kernel that reduces the computational load of the filtering process. Finally, a partial distortion search scheme is integrated into previous modifications to reduce the computational load without introducing any decrease in accuracy.

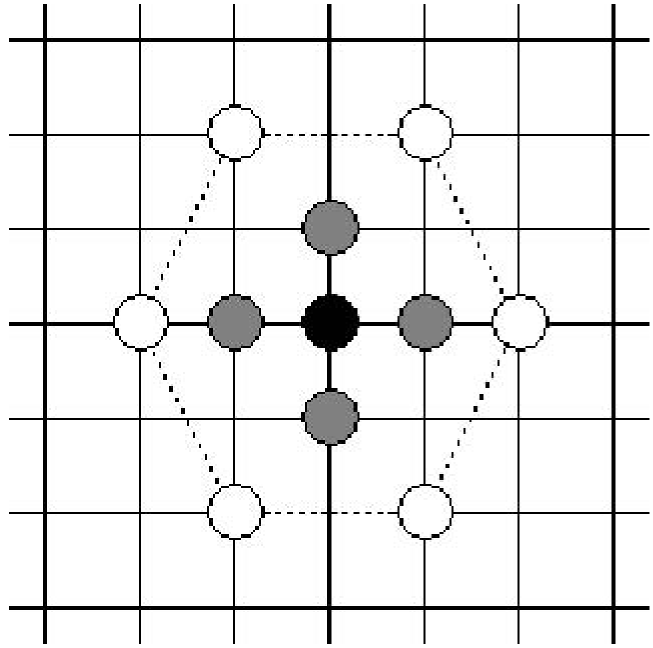

Fig. 1 Hexagonal search pattern.

The hexagonal search approach presented in Ref. 2 proposes using the search pattern given in Fig. 1. Initially, six hexagonal-shaped search points (white dots) around the search center (black dot) are checked. If the search center does not give the lowest error, the point giving the lowest error is set as the new search center. The search procedure continues with the new search center, and this process continues until the search center gives the lowest error. Next, four additional search points (gray dots) are checked, and the point providing the lowest error is assigned as motion vector. The utilization of spatial correlation between neighboring blocks (upper, left) together with the hexagonal search pattern is proposed in Ref. 5. This predictive hexagonal search pattern initially checks the zero motion case and the locations identified by the motion vectors of the upper and the left blocks. The motion vector giving the lowest error is chosen as the initial center point of hexagonal search. This modification not only reduces the average number of checking points but also increases the performance slightly since it exploits spatial correlation.

In the proposed method, the aforementioned predictive hexagonal search pattern is combined with the C-1BT-based motion estimation approach. As will be shown in the experimental result, this combination provides a significant reduction in the number of search points for sequences containing low motion without any loss in accuracy. However, it may cause an accuracy decrease for sequences including high motion, as for these cases the predictive hexagonal search pattern can be trapped in a suboptimal solution. Therefore, it is proposed to execute four additional hexagonal searches at the $(s/2,s/2)$, $(-s/2,s/2)$, $(s/2,-s/2)$, $(-s/2,-s/2)$ locations simultaneously with the original predictive hexagonal search, to improve accuracy at the expense of increasing the number of search points. Thus, five separate hexagonal searches are carried out at this stage. This trade-off will be discussed in Section 4.

Another modification proposed in this paper is to change the kernel used in the filtering process to speed up this procedure. The kernel employed in the convolution operation (i.e., filtering) for MF-1BT and C-1BT has a diamond shape and contains 16 nonzero components:

$$K_{MF\text{-}1\mathrm{BT}} = \frac{1}{16}\begin{bmatrix}
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0
\end{bmatrix}. \tag{5}$$

As is described in Ref. 8, the frequency response of this kernel is similar to the original 1BT kernel presented in Ref. 6. In this paper, the number of nonzero components in the filtering kernel is reduced to 4, as shown in (6), to enable faster implementation without changing the multiband-pass characteristic of the filter. Since the distance between nonzero elements of the kernel is increased in the new design, the number of passing bands is increased. However, as will be shown in Section 4 the performance of the proposed kernel is similar to the kernel used in the MF-1BT and C-1BT approaches.

$$K_{\mathrm{PROPOSED}} = \frac{1}{4}\begin{bmatrix}
0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0
\end{bmatrix}. \tag{6}$$

Only 4 additions and one shift operation are required for each pixel, as the proposed new kernel has only 4 nonzero components. These operations can be performed purely using integer arithmetic. It is possible to obtain binary one-bit images faster with the proposed kernel as the number of addition operations is reduced 4 times with respect to the C-1BT-based motion estimation method.

Partial distortion search (PDS) is a frequently used computational trick in block motion estimation. The main idea of partial distortion search is to discard impossible candidates if their accumulated error, up to the current computation, exceeds the minimum error already computed. In order to apply this trick to the proposed approach, we simply check the accumulated error after the error computation of each row and compare it to the current minimum error. Since this approach uses only a computational trick, this combination enables faster computation of the motion vector without any performance loss.

4 Experimental Results

Motion estimation has been performed for CIF-sized Foreman (300 frames) and Coastguard (300 frames) sequences as well as SIF-sized Garden (115 frames) and Mobile (140 frames) sequences using exhaustive full search with SAD matching, as well as 1BT as proposed in Ref. 6, 2BT as proposed in Ref. 7, MF-1BT as proposed in Ref. 8, and C-1BT as proposed in Ref. 9. Various cases are considered for the proposed scheme. First, the proposed kernel is combined with C-1BT to evaluate the effect of this new kernel (denoted as Case-1). Next, the combination of Case-1 with plain hexagonal search is tested to investigate the influence of the hexagonal search pattern (denoted as Case-2). Combination of the predictive hexagonal search pattern with Case-1 is examined to evaluate contribution of the correlation between neighboring blocks (denoted as Case-3). The Case-3 scheme is then unified with partial distortion search, and this combination is referred to as Case-4. Finally, the utilization of the additional four hexagonal searches for sequences containing high motion is assessed (denoted as Case-5).

Tables 1 and 2 show results of the compared methods for $s=16$. As PDS is used on a row-by-row basis, the number of average row computations using an XOR-based matching (NARC) criterion is utilized to compare the computational complexity of the motion estimation methods. The performance of these methods is evaluated using the peak signal-to-noise ratio (PSNR) measure that is computed between the original frames and frames reconstructed from the previous frame using motion vectors calculated from one of the methods. Note that computational savings obtained by means of the proposed kernel are not shown in these tables.

Results of Case-1 clearly confirm the effectiveness of the proposed kernel, as the PSNR values remain the same with respect to the C-1BT approach. When we evaluate the Case-2 approach, the computational gain in motion estimation is about 50 times. However, hexagonal search results in a significant accuracy decrease especially for the Foreman sequence, which contains high motion. Thanks to the predictive scheme, the Case-3 approach, on the other hand, not only reduces the computational load even further but also increases the accuracy considerably. Moreover, PSNR values of the Coastguard and Mobile sequences are slightly higher than the original C-1BT results in this case. In these situations, the best match for the C-1BT images does not exactly correlate to the best match in the image domain. Furthermore, in the proposed approach, the motion search is carried out in a region that is correlated to neighboring blocks; hence, the motion field is in some way explicitly related to the motion of neighboring blocks improving the motion estimation accuracy of C-1BT. Therefore, sometimes it may be that a suboptimal C-1BT match, found using the sparse search, actually is a better match in the image domain than the optimal C-1BT match.

Table 1 Performance comparison for the Coastguard and Foreman sequences.

	Coastguard			Foreman		
Method	PSNR (dB)	NARC	Gain (times)	PSNR (dB)	NARC	Gain (times)
FS-SAD	30.48	N/A	N/A	32.09	N/A	N/A
1BT [6]	29.83	13909.33	1	30.32	13909.33	1
2BT [7]	29.94	13909.33	1	30.70	13909.33	1
MF-1BT [8]	29.88	13909.33	1	30.38	13909.33	1
C-1BT [9]	29.98	13909.33	1	30.86	13909.33	1
Case-1	30.00	13909.33	1	30.88	13909.33	1
Case-2	29.97	262.34	53.02	29.36	319.42	43.55
Case-3	30.13	199.86	69.59	30.37	217.14	64.06
Case-4	30.13	84.67	164.27	30.37	118.68	117.20
Case-5	30.13	273.93	50.78	30.67	403.77	34.45

Table 2 Performance comparison for the Garden and Mobile sequences.

	Garden			Mobile		
Method	PSNR (dB)	NARC	Gain (times)	PSNR (dB)	NARC	Gain (times)
FS-SAD	23.79	N/A	N/A	22.99	N/A	N/A
1BT [6]	23.32	13751.27	1	22.71	13751.27	1
2BT [7]	23.43	13751.27	1	22.72	13751.27	1
MF-1BT [8]	23.29	13751.27	1	22.73	13751.27	1
C-1BT [9]	23.39	13751.27	1	22.77	13751.27	1
Case-1	23.37	13751.27	1	22.79	13751.27	1
Case-2	22.94	274.93	50.02	22.53	317.91	43.26
Case-3	23.39	205.10	67.05	22.80	197.01	69.80
Case-4	23.39	133.29	103.17	22.80	89.69	153.32
Case-5	23.37	375.40	36.63	22.81	322.15	42.69

The Case-4 approach saves significant computations while keeping performance at a similar level with Case-3. The computational load can be decreased over 100 times in total using this approach. Case-3 and Case-4 approaches give lower performance for the Foreman sequence. The main reason for this is that hexagonal-based search is not very effective to estimate large motions when it is combined with C-1BT-based motion estimation. This drawback can be alleviated using the aforementioned four additional hexagonal searches at the cost of some computational load (Case-5). It may also be possible to completely prevent this shortcoming using a full-search scheme when the matching is not sufficient. This can be done by simply checking the CNNMP measure with a predefined or adaptive threshold at the expense of some more computation. Overall, the proposed approach provides significant reduction of computational load and outperforms 1BT-,[6] 2BT-,[7] and MF-1BT,[8] based motion estimation approaches.

5 Conclusions

In this paper, the C-1BT-based motion estimation approach is combined with a predictive hexagonal search pattern to reduce the computational load of motion estimation. Additionally, PDS is also integrated into the proposed scheme to further speed up the motion estimation procedure. Furthermore, the novel kernel proposed reduces the computational load of the filtering. Experimental results present a significant reduction of computational load with nearly similar accuracy.

Acknowledgments

The author would like to thank the anonymous reviewers for their insightful and constructive recommendations, which have improved the understanding and representation of the paper.

References

1. S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.* **9**(2), 287–290 (2000).
2. C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* **12**(5), 349–355 (2002).
3. S. G. Park and D. S. Jeong, "Fast adaptive diamond search algorithm for block-matching motion estimation using spatial correlation." In Visual Commun. Image Process., *Proc. SPIE* **4310**, 363–371 (2001).
4. J. Y. Kim and S. B. Yang, "Efficient search algorithm for block motion estimation," *J. Electron. Imaging* **11**(1), 69–74 (2002).
5. C. Zhu, X. Lin, L. P. Chau, and L. M. Po, "Enhanced hexagonal search for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* **14**(10), 1210–1214 (2004).
6. B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *IEEE Trans. Circuits Syst. Video Technol.* **7**(4), 702–706 (1997).
7. A. Ertürk and S. Ertürk, "Two-bit transform for binary block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* **15**(7), 938–946 (2005).
8. S. Ertürk, "Multiplication-free one-bit transform for low-complexity block-based motion estimation," *IEEE Signal Process. Lett.* **14**(2), 109–112 (2007).
9. O. Urhan and S. Ertürk, "Constrained one-bit transform for low-complexity block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* **17**(4), 478–482 (2007).
10. P. H. W. Wong and O. C. Au, "Modified one-bit transform for motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* **9**(7), 1020–1024 (1999).
11. B. Demir and S. Ertürk, "Block motion estimation using modified two bit transform," *Lect. Notes Comput. Sci.* **4263**, 522–531 (2006).
12. Y. K. Wang and G. F. Tu, "Successive elimination algorithm for binary block matching motion estimation," *Electron. Lett.* **36**(24), 2007–2008 (2000).

Oğuzhan Urhan received his BSc, MSc, and PhD degrees in electronics and telecommunications 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 appointed as assistant professor. He worked as a visiting professor at Chung-Ang University, Korea, between September 2006 and March 2007. His research interests are in the area of digital signal and image processing, in particular, image and video restoration and coding.