Prediction-based Directional Search for Fast Block-Matching Motion Estimation

Binh P. Nguyen
School of Information and Communication Technology, Hanoi University of Technology, Vietnam
binhnp@it-hut.edu.vn

Trang T. T. Do
Department of Electrical and Computer Engineering, National University of Singapore, Singapore
dothutrang@nus.edu.sg

Chee-Kong Chui
Department of Mechanical Engineering, National University of Singapore, Singapore
mpecck@nus.edu.sg

Sim-Heng Ong
Department of Electrical and Computer Engineering, National University of Singapore, Singapore
eleosh@nus.edu.sg

ABSTRACT

This paper proposes an efficient block-matching motion estimation algorithm known as *prediction-based directional search* (PDS). This new algorithm is applicable to a wide range of video processing applications. The algorithm uses the motion vectors in two neighboring blocks to predict a starting search point for the current block. The subsequent refining search relies on the hypothesis of monotonic block distortion surface and the center-biased characteristic of motion vector probability distribution. The cross pattern in a step and one of four possible directional rectangle search patterns in the next step are iteratively used to find the motion vector. Experiments on eleven video sequences with different characteristics shows that PDS can achieve a faster computation speed with similar or even better distortion performance compared to some existing well-known algorithms.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Miscellaneous

General Terms

Algorithms

Keywords

PDS, prediction, directional search, block-matching, motion estimation

1. INTRODUCTION

In video coding, motion estimation plays an important role since it is used to efficiently reduce the temporal redundancy information between successive frames, thus improving the compression ratio. Among existing motion estimation methods, block-matching algorithms (BMAs) are widely used because of their simplicity and effectiveness. Block-matching aims to find, within a search window, the best-matched block from the previous frame based on a block distortion measure or other matching criteria. The displacement of the best-matched block is described as a motion vector (MV) to the block in the current frame. The algorithm that can find the best-matched block is the full search (FS) method, which evaluates all the candidate blocks within the search window. However, the high computational cost of FS limits its application in practice. In order to reduce the heavy computational complexity of FS, numerous fast BMAs have been proposed. Most fast BMAs exploit different search patterns to reduce the number of search points when finding the best-matched block, such as square patterns in the three-step search (TSS) [8], new three-step search (NTSS) [9], efficient three-step search (E3SS) [6], four-step search (4SS) [13], and block-based gradient descent search (BBGDS) [10]; cross patterns in the cross search algorithm (CSA) [3] and two-dimensional logarithmic search (TDLS) [5]; diamond patterns in diamond search (DS) [17] and cross-diamond search (CDS) [2]; and hexagon patterns in hexagon-based search (HEXBS) [16]. The primary assumption of these fast BMAs is that the block distortion decreases monotonically when the search position moves toward the minimum matching distortion point (MMDP). Hence, it is not necessary to check all the search points in the search window since the best-matched position can be found by tracking the changing trend of the distortion. In addition, since there is a high correlation between the current block and its neighboring blocks in spatial and temporal domains, the current block's MV can be predicted based on the MVs of the neighboring blocks. The predicted MV (PMV) can be obtained by calculating the mean, median, weighted mean or other statistical average values of the neighboring MVs [11, 15]. It also can be selected as one of the neighboring MVs based on a certain criteria [4, 14, 7, 12]. After the prediction, a pattern-based searching method or the FS with redefined searching area is performed around the position corresponding to the PMV to determine the final MV. Although addition memory is needed to store the neighboring MVs, these prediction-based BMAs often offer a better performance compared to conventional BMAs. It is due to the fact that the prediction step can find a good starting point for the refining search. This can reduce the unnecessary intermediate search incurred when applying large search patterns to slow motion scenes or the risk of being trapped in a local MMDP along the search path when searching for a large MV.

In this paper, we propose a novel BMA using prediction, called *prediction-based directional search* (PDS). Our algorithm relies on a simple and effective strategy to predict the starting search point with high probability to be close to the global MMDP, and the two small and compact search patterns which are then applied to quickly direct the search position moving to the MMDP. Experimental results show that our algorithm can achieve substantial speed improvement over other fast BMAs with similar distortion performance.

The rest of the paper is organized as follows. The strategy to predict the starting search point for PDS is presented in Section 2. Section 3 describes the search patterns used in PDS. The full description of the new algorithm is provided in Section 4. Our experiments are described and discussed in Section 5. The paper concludes in Section 6.

2. PREDICTION OF STARTING SEARCH POINT

Apparently, the high correlation characteristic between the current block and its neighboring blocks in both spatial and temporal domains can be exploited to predict the MV of the current block. However, utilizing temporal correlation might be undesirable in practical since the entire MVs of the reference frame needed to be recorded. In spatial domain, since all the blocks in the same frame are processed in a raster scan order, the neighboring blocks having MVs which can be used for the prediction are on the left, above-left, above, and above-right directions of the current block. They are denoted as block A, B, C, and D, respectively, as illustrated in Figure 1. There is a compromise between the accuracy of the prediction and the computational complexity based on the number of the neighboring blocks to be evaluated. Block B is rarely used due to its high correlation with block A and C. The international video coding standard H.263 uses block A, C, and D for differential coding of MVs. These three blocks also were used in [11] to predict the MV of the current block. Xu *et al.* used only block A for the same purpose [15]. Since in real-world video sequences, the movements in horizontal and vertical directions are dominant, block A and C will be used in our method.

In our proposed algorithm, the PMV of the current block is obtained by evaluating the block distortion values at the three points determined by vector (0, 0) and the MVs of the left and above neighboring blocks, i.e. block A and C. The MMDP in this step is then chosen as the starting search point for subsequent steps. It is noted that if both block A

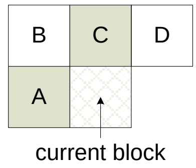


Figure 1: Blocks can be exploited to predict the MV. Shaded blocks are used in the proposed algorithm.

and C are not available, this prediction step will be skipped and the refining search will start from the point (0, 0). In the case that only one of these two blocks is available, the MV of the other and vector (0, 0) will be used to predict the PMV.

3. SELECTION OF SEARCH PATTERNS

Previous studies reported that the probability of MV in horizontal and vertical directions is higher than that in other directions [3, 17, 2]. It may be due to the fact that most of the object and camera movements are in these directions. We exploit this center-biased characteristic of motion vector probability distribution in our algorithm by first applying a small cross search pattern at the starting search point determined by the PVM. In addition, our method also relies on the hypothesis of monotonic block distortion surface. This implies that the MMDP can be found along the directions of the distortion from the higher points to the lowest point. Four directional search patterns as depicted in Figure 2 are chosen to utilize this assumption. The specific directional search pattern used in current step is selected based on the position of the MMDP in the cross search pattern applied to the previous step. Our preliminary experimental results showed that these compact search patterns can lead the search position to move quickly to the final MMDP.

4. NEW PREDICTION-BASED MOTION ESTIMATION

The algorithm comprises the following steps:

- Step 1: The center of the search window for the current block, i.e. the starting search point, is selected as the MMDP when evaluating the three points determined by vector (0, 0) and the MVs of the left and above neighboring blocks.
- Step 2: Apply the cross search pattern at the center of the search window. Evaluate the matching distortion of all search points in this pattern. If the MMDP occurs at the center, the search process stops and the motion vector is found at the center; otherwise, go to step 3.
- Step 3: Based on the position of the MMDP found in step 2, choose a proper directional search pattern accordingly to find the new MMDP. If the location of the MMDP remains unchanged, the search process stops with the motion vector found at the MMDP. If not, go back to step 2.

Figure 3 is an example of PDS in which the motion vector (1, 4) is obtained after evaluating 19 search points in 5 steps.

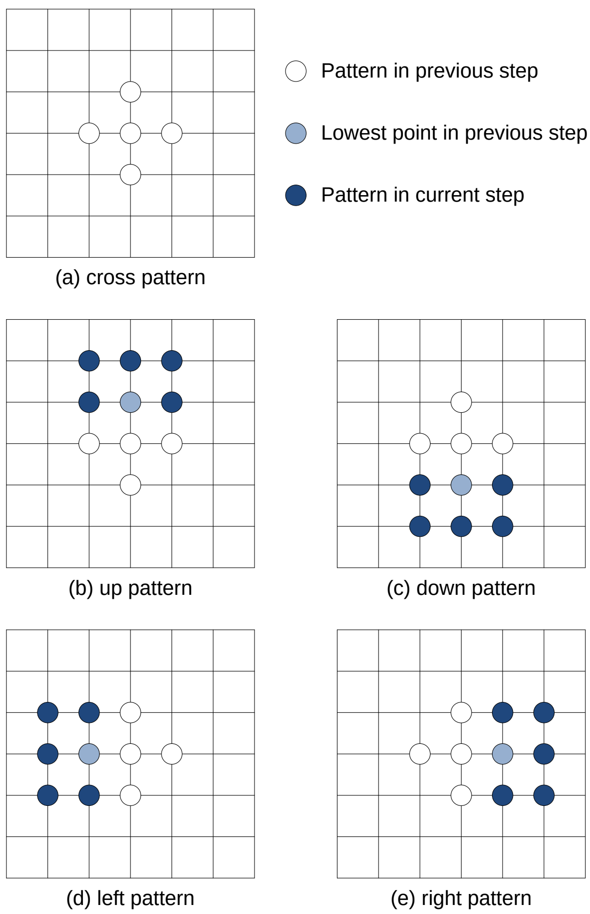


Figure 2: Search patterns of PDS.

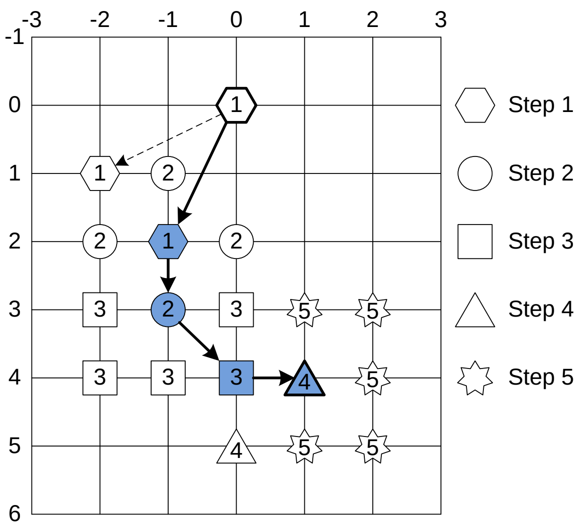


Figure 3: An example of PDS.

In step 1, i.e. starting search point prediction, the PMV (-1, 2) is determined after checking 3 search points. In step 2, four search points in the cross pattern centered at (-1, 2) are checked. Subsequently, there are five more points to be checked in step 3. From step 4 onwards, only a few points in the search patterns are evaluated since some of them have already been checked.

5. EXPERIMENTAL RESULTS AND DISCUSSION

In motion estimation for video coding, the typical block distortion measures include mean squared error (MSE), mean absolute error (MAE), and sum of absolute differences (SAD). Although both the MAE and SAD operations can be viewed as equivalent, additional shifts or divides may be needed to obtain the MAE. Calculating the MSE requires numerous multiplication, however, it does not produce any significant improvement of the performance compared to the two other measures. In this regard, SAD was used as the block distortion measure in our simulations. The block size was chosen as of 16×16 and the search window size was 15×15 pixels. This means that each component of the motion vector ranges from -7 to $+7$. Eleven video sequences with difference motion speeds, frame sizes, frame rates, and durations were used in the experiments. The descriptions of these datasets are given in Table 1 and 2.

In addition to FS, eight fast BMAs including NTSS, E3SS, 4SS, DS, CDS, HEXBS, prediction search algorithm (PSA) [11], and adative rood pattern search (ARPS) [12] were implemented to compare their performances with the proposed algorithm. It is noted that PSA, ARPS, and our proposed PDS are predict-based methods. Three criteria were used for the comparisons: the average of search points per block, the speed-up ratio with respect to FS, and the average of peak signal to noise ratio (PSNR) in dB between predicted frames and original frames. Table 3, 4, and 5 summarize the experimental results.

As can be seen from these tables, the proposed PDS out performs all other algorithms in term of speed which was measured by the number of search points. The speed-up ratio with respect to FS of PDS ranges from about 13 to 33. PDS runs with the speed of almost two times faster than DS, a well-known BMA which was accepted in the MPEG-4 Verification Model (VM) [1]. In addition, PDS is the best algorithm in term of fidelity when applied to video sequences with background changing and fast motion content, e.g. `Football`, `Running`, and `Waves`. For other sequences, PDS achieved an above-average PSNR compared to other algorithms. PDS also produces higher PSNR values compared to the second fastest algorithm, ARPS, when applied to almost all the test video sequences.

Figure 4 illustrates the frame by frame comparison of PSNR and the average of search points per block when applying all fast BMA algorithms in our experiments to the `Tourists` sequence. It clearly demonstrates the robustness of the proposed PDS to other algorithms in term of both speed and fidelity.

6. CONCLUSION

We have proposed an efficient prediction-based BMA which can be applied to a wide range of video applications. The strategy to predict the motion vector in the algorithm can

Table 1: Specifications of the test video sequences

Sequence	Frame size (pixels)	Frame rate (fps)	Number of frame
`Baby`	640 × 480	30.00	300
`Boat`	384 × 288	25.00	200
`Coral`	512 × 288	25.00	300
`Dancing`	528 × 304	25.00	300
`Dolphins`	384 × 288	25.00	300
`Football`	640 × 368	29.97	300
`Island`	384 × 288	25.00	100
`Riding`	608 × 256	23.98	95
`Running`	576 × 240	25.00	300
`Tourists`	384 × 288	25.00	180
`Waves`	384 × 288	25.00	300

Table 2: Characteristics of the test video sequences

Sequence	Motion content	Background changing	Camera zooming	Camera panning
`Baby`	Normal	No	No	No
`Boat`	Slow	Slow	No	No
`Coral`	Slow	Slow	Slow	Slow
`Dancing`	Fast	Slow	No	Slow
`Dolphins`	Fast	Fast	No	Fast
`Football`	Fast	Normal	Slow	Slow
`Island`	Slow	Slow	Slow	Slow
`Riding`	Fast	Fast	No	Fast
`Running`	Fast	Fast	Slow	Normal
`Tourists`	Normal	Normal	Normal	No
`Waves`	Fast	Normal	No	Normal

Table 3: Average of search points per block

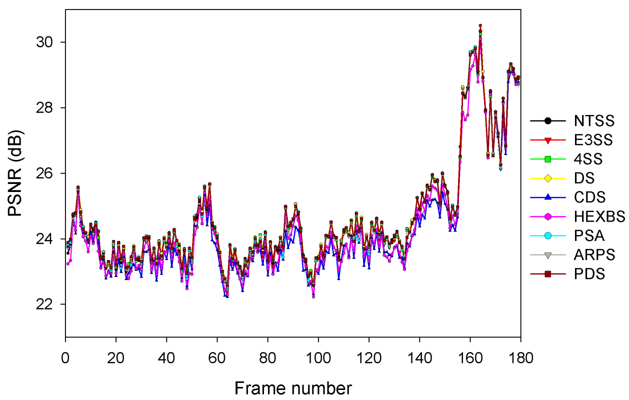
	Baby	Boat	Coral	Dancing	Dolphins	Football	Island	Riding	Running	Tourists	Waves
FS	212.91	205.04	207.11	207.90	205.04	210.83	205.04	206.67	205.53	205.04	205.04
NTSS	20.48	18.10	19.70	18.79	28.05	20.01	19.31	28.43	26.60	19.62	24.04
E3SS	17.13	14.73	16.67	15.40	25.17	16.76	16.43	25.45	24.34	16.89	22.07
4SS	18.40	16.93	17.93	17.36	22.67	17.89	18.07	22.36	21.71	17.96	20.35
DS	15.63	13.92	15.28	14.53	24.03	15.45	15.28	23.47	21.94	15.32	19.91
CDS	13.80	10.85	13.19	11.75	27.82	13.65	13.92	28.37	25.70	13.67	22.59
HEXBS	12.38	11.31	12.13	11.69	16.09	12.25	12.32	16.50	15.82	12.23	14.72
PSA	12.87	10.84	11.64	11.25	23.43	11.92	11.28	20.85	17.97	11.47	13.72
ARPS	8.47	7.48	8.17	7.69	15.59	8.00	8.41	12.97	11.49	8.54	10.22
PDS	7.76	6.54	6.26	6.57	15.14	6.66	6.59	11.49	8.44	6.41	7.92

Table 4: Speed-up ratio with respect to FS

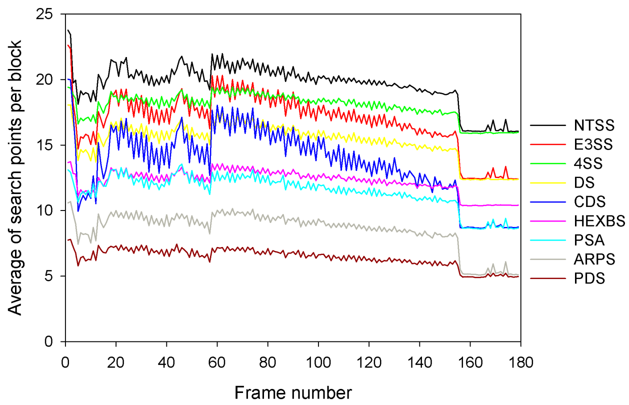
	Baby	Boat	Coral	Dancing	Dolphins	Football	Island	Riding	Running	Tourists	Waves
FS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
NTSS	10.40	11.33	10.51	11.06	7.31	10.54	10.62	7.27	7.73	10.45	8.53
E3SS	12.43	13.92	12.42	13.50	8.15	12.58	12.48	8.12	8.44	12.14	9.29
4SS	11.57	12.11	11.55	11.98	9.04	11.78	11.35	9.24	9.47	11.42	10.08
DS	13.62	14.73	13.55	14.31	8.53	13.65	13.42	8.81	9.37	13.38	10.30
CDS	15.43	18.90	15.70	17.69	7.37	15.45	14.73	7.28	8.00	15.00	9.08
HEXBS	17.20	18.13	17.07	17.78	12.74	17.21	16.64	12.53	12.99	16.77	13.93
PSA	16.54	18.92	17.79	18.48	8.75	17.69	18.18	9.91	11.44	17.88	14.94
ARPS	25.14	27.41	25.35	27.04	13.15	26.35	24.38	15.93	17.89	24.01	20.06
PDS	27.44	31.35	33.08	31.64	13.54	31.66	31.11	17.99	24.35	31.99	25.89

Table 5: Average of PSNR (dB) per frame. The highest values in each column apart from FS are underlined

	Baby	Boat	Coral	Dancing	Dolphins	Football	Island	Riding	Running	Tourists	Waves
FS	35.88	38.74	31.57	30.71	25.72	33.18	34.70	23.74	29.30	24.74	31.92
NTSS	35.79	38.67	31.51	30.55	25.52	33.11	34.64	23.47	28.62	24.64	31.22
E3SS	35.77	38.59	31.50	30.55	25.54	33.11	34.60	23.49	28.69	24.62	31.63
4SS	35.76	38.09	31.53	30.50	25.42	33.01	34.06	23.23	28.30	24.64	31.47
DS	35.79	38.55	31.55	30.51	25.39	33.04	34.68	23.17	28.34	24.67	31.57
CDS	35.69	38.16	31.11	30.37	25.29	32.98	23.10	34.14	28.09	24.29	31.47
HEXBS	35.62	38.10	31.11	30.35	25.29	32.87	33.19	23.07	28.05	24.33	31.15
PSA	35.76	38.62	31.55	30.50	25.17	32.48	34.68	23.15	28.58	24.63	31.73
ARPS	35.74	38.57	31.51	30.45	25.41	33.12	34.66	23.56	28.86	24.61	31.83
PDS	35.76	38.58	31.52	30.47	25.28	33.13	34.68	23.48	28.93	24.65	31.86



(a)



(b)

Figure 4: Frame by frame comparisons for Tourists sequence: (a) Average of PSNR per frame; (b) Average search points per block.

effectively move the starting search point close to the global MMDP. This can reduce the risk of being trapped in a local MMDP along the search path when searching for a large MV and avoid unnecessary intermediate search incurred when processing slow motion scenes. Our directional search patterns also help the subsequent refining search can quickly find the final MMDP. Experimental results showed that our algorithm can achieve a faster computation speed with similar or even better distortion performance compared to other well-known BMAs when applied to video sequences with various characteristics.

7. REFERENCES

[1] Mpeg-4 video verification model version 14.0. In S. Fukunaga, Y. Nakaya, S.-H. Son, and T. Nagumo, editors, *Coding of moving pictures and audio.* ISO/IEC JTC1/SC29/WG11 N2932, Melbourne, Oct. 1999.

[2] C.-H. Cheung and L.-M. Po. A novel cross-diamond search algorithm for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:1168–1177, 2002.

[3] M. Ghanbari. The cross-search algorithm for motion estimation. *IEEE Transactions on Communications*, 38:950–953, 1990.

[4] C.-H. Hsieh, P.-C. Lu, J.-S. Shyn, and E.-H. Lu. Motion estimation algorithm using interblock correlation. *Electronics Letters*, 26(5):276–277, 1990.

[5] J. Jain and A. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, 29:1799–1808, 1981.

[6] X. Jing and L.-P. Chau. An efficient three-step search algorithm for block motion estimation. *IEEE Transactions on Multimedia*, 6:435–438, 2004.

[7] D.-W. Kim, J.-S. Choi, and J.-T. Kim. Adaptive motion estimation based on spatio-temporal correlation. *Signal Processing: Image Communication*, 13(2):161–170, 1998.

[8] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro. Motion compensated interframe coding for video conferencing. In *Proceedings of the National Telecommunications Conference*, pages G5.3.1–G5.3.5, 1981.

[9] R. Li, B. Zeng, and M.-L. Liou. A new three-step search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 4:438–442, 1994.

[10] L.-K. Liu and E. Feig. A block-based gradient descent search algorithm for block motion estimation in video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:419–422, 1996.

[11] L. Luo, C. Zou, X. Gao, and Z. He. A new prediction search algorithm for block motion estimation in video coding. *IEEE Transactions on Consumer Electronics*, 43(1):56–61, 1997.

[12] Y. Nie and K.-K. Ma. Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, 11(12):1442–1449, 2002.

[13] L.-M. Po and W.-C. Ma. A novel four-step search algorithm for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:313–317, 1996.

[14] J.-C. Tsai, C.-H. Hsieh, S.-K. Weng, and M.-F. Lai. Block-matching motion estimation using correlation search algorithm. *Signal Processing: Image Communication*, 13(2):119–133, 1998.

[15] J.-B. Xu, L.-M. Po, and C.-K. Cheung. Adaptive motion tracking block matching algorithms for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1025–1029, 1999.

[16] C. Zhu, X. Lin, and L.-P. Chau. Hexagon-based search pattern for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:349–355, 2002.

[17] S. Zhu and K.-K. Ma. A new diamond search algorithm for fast block matching motion estimation. In *Proceedings of the International Conference on Information, Communications and Signal Processing 1997 (ICICS'97)*, volume 1, pages 292–296, 1991.