J Real-Time Image Proc
DOI 10.1007/s11554-012-0264-7

ORIGINAL RESEARCH PAPER

Predictive and distribution-oriented fast motion estimation for H.264/AVC

Zhaoqing Pan · Sam Kwong · Long Xu · Yun Zhang · Tiesong Zhao

Received: 19 March 2012 / Accepted: 25 June 2012

Abstract For fast motion estimation (ME) in video coding, many fast block matching ME algorithms are proposed. Among these algorithms, Unsymmetrical-cross Multi-grid-hexagon Search (UMHexagonS) algorithm can be regarded as a distinguished representative. However, the excellent rate-distortion (R-D) performance of UMHexagonS comes at the cost of relatively high computational complexity of the initial search point decision and the hybrid search pattern. To tackle this disadvantage, a new fast ME algorithm is proposed. An experiment is performed to analyze the best motion vectors (MVs) distribution in natural video sequences. Based on the correlations between spatial and temporal blocks as well as the asymmetrical distribution of the best MVs in natural video sequences, a small diamond search pattern and an asymmetrical cross search pattern are jointly employed to locate the best matching block. Experimental results demonstrate that when compared to recently improved UMHexagonS, the ME time can be reduced up to 38.70 % while with a quite similar R-D performance as UMHexagonS. When compared with the fast directional gradient descent search (FDGDS), the ME time can be reduced up to 12.23 %, while with a better R-D performance than FDGDS, 0.11 dB BDPSNR increase and 2.14 % BDBitrate decrease. Especially, the proposed algorithm can work well in video sequences with various motion activities and formats, and is more suitable for real-time application.

Keywords Block matching algorithm · Asymmetrical cross search pattern · Motion estimation · H.264/AVC · Video coding

Z. Pan · S. Kwong (✉) · L. Xu · Y. Zhang · T. Zhao
Department of Computer Science,
City University of Hong Kong, Hong Kong, China
e-mail: cssamk@cityu.edu.hk

Z. Pan
e-mail: zqpan3@student.cityu.edu.hk

L. Xu
e-mail: lxu@jdl.ac.cn

Y. Zhang
e-mail: yunzhang@cityu.edu.hk

T. Zhao
e-mail: tiesong.zhao@cityu.edu.hk

Y. Zhang
Shenzhen Institutes of Advanced Technology,
Chinese Academy of Sciences, Shenzhen, China

1 Introduction

H.264/AVC (advanced video coding) is the state-of-the-art standard for video coding [1, 2]. Since H.264/AVC utilizes a set of advanced video coding tools, such as variable block-size (16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4) ME, multiple reference frames [3], R-D optimization technique and so on, it significantly improves the coding performance when compared with previous video coding standards. However, the improvement is at the cost of higher computational complexity. In [4], it has been proved that the variable block-size ME consumes 70 % (one reference frame) $\sim$90 % (five reference frames) of the total encoding time of the H.264/AVC encoder. This high computational burden limits the use of H.264/AVC in real-time video coding.

In order to reduce the computational complexity of ME process, many fast block matching ME algorithms have been proposed. Based on the assumption that the block matching error decreases monotonically in the search window, these algorithms employ different search patterns

Published online: 24 August 2012

(such as square search pattern [5–7], diamond search pattern [8, 9], hexagon search pattern [9, 10, 20] and so on) to locate the best matching block. To further reduce the computational complexity of the block matching process, different search steps (like three steps [5], four steps [6] and so on) are utilized to reduce the number of search points. In [5] and [6], a square search pattern is used with three steps and four steps, respectively. They are named three-step search (TSS) and four-step search (FSS). The initial search step size of TSS and FSS are corresponding to $\lceil w/2 \rceil$ and $\lceil w/4 \rceil$, respectively, where $\lceil \cdot \rceil$ denotes the ceiling operation. w represents the search range. In [7], a square search pattern with eight points is performed on the initial search point. If the best point is found, then the same square search pattern is performed on the best point. This process is recursively implemented until the best point locates at the center of the square search pattern. In [8], a large diamond search pattern (LDSP) with eight search points and a small diamond search pattern (SDSP) with four search points are employed. The LDSP is recursively implemented until the best search point is the center of LDSP, then an SDSP is used to refine the search result. In [10], a hexagon-based block matching algorithm is proposed. Firstly, it searches a relatively best point by recursively implementing the hexagon search pattern (HSP). Then an SDSP is additionally employed to refine the search result. These algorithms are presented based on the unimodal error surface assumption [11]. However, this assumption may result in a local minimum. In order to solve the local minimum problem, UMHexagonS [12, 13] is proposed and adopted in H.264/AVC reference software. It takes advantages of multiple initial search point decision scheme and hybrid search pattern. However, the excellent R-D performance comes at the high computational complexity of these two techniques.

In this paper, we propose a new efficient block matching algorithm. First of all, according to the spatial and temporal correlations in natural video sequences, three predictive models are jointly employed to determine a more accurate initial search point. After getting the initial search point, according to the center-biased concept of block matching as well as the asymmetrical characteristics of the best motion vectors (MVs) in natural video sequences, an SDSP and an asymmetrical cross search pattern (ACSP) are employed to locate the best matching block. The experimental results demonstrate that the proposed algorithm yields a quite promising performance in terms of R-D performance and computational complexity. Especially, the proposed algorithm is more suitable for real-time application.

The rest of this paper is organized as follows. The details of the proposed fast ME algorithm are described in Sect. 2. Experimental results are shown in Sect. 3. At last, Sect. 4 concludes this paper.

2 Proposed fast ME algorithm

In this section, two techniques that are jointly used to locate the best matching block are introduced. One is the initial search point decision scheme. In order to take full advantage of the correlations between spatial and temporal blocks, three predictive models are jointly employed to determine an initial search point. The other is the best block search strategy. An experiment is performed to analyze the best MVs distribution in natural video sequences. According to the distribution, different block matching search patterns are used to locate the best matching block.

2.1 Initial search point decision

To avoid the disadvantages of unimodal error surface assumption [11] which may lead block matching search into a local minimum, we should make full use of correlations in natural video sequences. Only in this way can we get a more accurate initial search point. It is well known that there are strong correlations between MVs in both spatial and temporal neighboring blocks in natural video sequence. In spatial domain, the correlations exist in the neighboring macroblocks and inner neighboring blocks of one Macroblock (MB). In temporal domain, there is a strong correlation between the collocated blocks of two consecutive frames. In order to take full advantages of these correlations, three conventional predictive models are used to locate the initial search point: median-predictive model (MPM), inner neighboring block-predictive model (INBPM) and collocated block-predictive model (CBPM) [12–14], they are corresponding to the neighboring MB correlation, inner neighboring block correlation and collocated block correlation, respectively. Since the spatial neighboring blocks on the top, top-right, and left of the current block have been encoded, MPM uses the median value of these three spatial neighboring blocks as the initial search point candidate, and is defined as

$$\mathrm{MV_{MPM}} = \mathrm{median}\{\mathrm{MV_T}, \mathrm{MV_{TR}}, \mathrm{MV_L}\}, \tag{1}$$

where $\mathrm{MV_T}$, $\mathrm{MV_{TR}}$, $\mathrm{MV_L}$ denote the MV of top, top-right and left block of the current block, respectively, as shown in Fig. 1. Because H.264/AVC supports seven block-size ME, the MV of current block size is highly correlated with its inner neighboring block size, for example, the neighboring block size of 16×8 and 8×16 are 16×16, the neighboring block size of 8×8 is 16×8 or 8×16, the neighboring block size of 8×4 and 4×8 are 8×8, and so on. Based on this characteristic, the MV of neighboring block size is employed by INBPM, and the predictive result is set as the initial search point candidate. In natural video sequences, the motion of one object is successive until the scene change happens. CBPM utilizes the MV of collocated block in previous frame as the initial search point candidate, Fig. 2 gives an illustration on CBPM. Finally, the initial search point is selected from these three MV candidates as Eq. (2).

$$\mathrm{MV}_{\text{initial}} = \min\{J(\mathrm{MV}_{\text{candidate}}, \lambda_{\text{MOTION}}), \mathrm{MV}_{\text{candidate}} \in \{\mathrm{MV}_{\text{MPM}}, \mathrm{MV}_{\text{INBPM}}, \mathrm{MV}_{\text{CBPM}}\}\}, \quad (2)$$

where J represents the R-D cost function [15]. MV_{MPM}, $\mathrm{MV}_{\text{INBPM}}$ and $\mathrm{MV}_{\text{CBPM}}$ denote the MVs are derived by MPM, INBPM and CBPM, respectively.

2.2 Distribution-based block matching search patterns

In fast ME algorithms, the block matching search pattern is highly concerned with the coding efficiency in terms of computational complexity and R-D performance. For exploiting the characteristics of the best MVs distribution in natural video sequences, an experiment is performed on the standard video sequences with JM14.1 reference codec of H.264/AVC [16]. The test conditions are listed in Table 1, the search range is set to 32 and the full search (FS) ME algorithm is used. The statistical results for three QCIF, three CIF and three HD (720P) video sequences are listed in Tables 2, 3, 4 and 5 based on different Q_p values. Table 6 shows the summary results.

In Tables 2, 3, 4, 5 and 6, **(0, 0)** represents the initial search point which is selected as the best point finally. **(X, 0)** denotes the best MVs which locate at the horizontal direction of the initial search point. **(0, Y)** represents the best MVs locate at the vertical direction of the initial search point. **(X, Y)** is the other conditions.

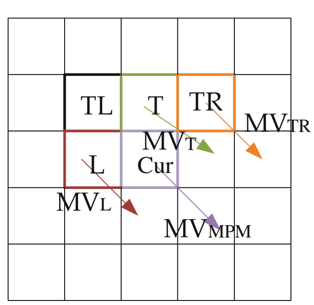


Fig. 1 Illustration of median-predictive model

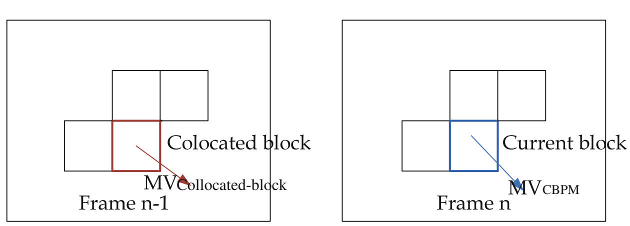


Fig. 2 Illustration of collocated block-predictive model

Table 1 Test conditions

Profile	Main
Quantization parameter (Q_p)	24, 28, 32, 36
Subpel ME	Enable
RD optimization	High complexity mode
Number of reference frames	5
GOP structure	IBPBP
Symbol mode	CABAC

Table 2 The best MVs distribution found by FS, $Q_p = 24$

Format	Sequence	**(0, 0)** (%)	**(X, 0)** (%)	**(0, Y)** (%)	**(X, Y)** (%)
QCIF	Container	87.04	10.39	1.65	0.92
	Mobile	40.77	18.14	4.95	36.14
	Salesman	89.38	2.55	2.38	5.69
CIF	Bus	16.70	16.74	5.11	61.45
	News	84.75	5.87	3.58	5.80
	Mother	72.52	7.94	4.58	14.96
HD (720P)	Harbour	26.07	23.94	11.15	38.84
	Shields	37.31	46.17	2.77	13.75
	Stockholm	35.26	44.67	3.11	16.96
Average		54.42	19.60	4.36	21.62

Table 3 The best MVs distribution found by FS, $Q_p = 28$

Format	Sequence	**(0, 0)** (%)	**(X, 0)** (%)	**(0, Y)** (%)	**(X, Y)** (%)
QCIF	Container	90.90	8.08	0.78	0.24
	Mobile	42.65	18.06	4.83	34.46
	Salesman	90.02	2.54	2.66	4.78
CIF	Bus	18.44	18.95	6.81	55.80
	News	86.25	5.68	3.21	4.86
	Mother	75.97	7.91	4.66	11.46
HD (720P)	Harbour	30.56	27.72	10.08	31.64
	Shields	41.80	47.22	1.98	9.00
	Stockholm	47.09	41.04	2.98	8.89
Average		58.19	19.69	4.22	17.90

From Tables 2, 3, 4 and 5, it can be observed that with the increase of Q_p, the ratio of **(0, 0)** increases, while the ratio of **(X, Y)** decreases, but the ratio of **(X, 0)** and **(0 ,Y)** almost remains unchanged. Another observation from the tables, video sequences with slow motion or simple context (such as *Container*, *Salesman*), **(0, 0)** would be selected as the best MV at the probability of 90 %. For fast motion or complex texture video sequences (such as *bus*, *mobile*), the percentage of **(X, Y)** is larger than that of the slow and simple context sequences.

Table 4 The best MVs distribution found by FS, $Q_p = 32$

Format	Sequence	(**0**, **0**) (%)	(**X**, **0**) (%)	(**0**, **Y**) (%)	(**X**, **Y**) (%)
QCIF	Container	92.72	7.00	0.24	0.04
	Mobile	44.91	17.92	5.04	32.13
	Salesman	90.89	2.48	2.80	3.83
CIF	Bus	22.28	21.28	8.33	48.11
	News	87.66	5.61	2.86	3.87
	Mother	80.18	7.34	4.38	8.10
HD (720P)	Harbour	35.74	31.65	8.57	24.04
	Shields	45.76	45.25	2.10	6.89
	Stockholm	52.25	38.88	2.97	5.90
Average		61.38	19.71	4.14	14.77

Table 5 The best MVs distribution found by FS, $Q_p = 36$

Format	Sequence	(**0**, **0**) (%)	(**X**, **0**) (%)	(**0**, **Y**) (%)	(**X**, **Y**) (%)
QCIF	Container	94.10	5.83	0.06	0.01
	Mobile	47.98	17.22	5.68	29.12
	Salesman	92.25	2.33	2.66	2.76
CIF	Bus	28.48	23.50	9.00	39.02
	News	89.21	5.35	2.45	2.99
	Mother	84.79	6.18	3.35	5.68
HD (720P)	Harbour	42.96	32.59	7.22	17.23
	Shields	50.84	41.90	2.25	5.01
	Stockholm	57.83	35.36	2.76	4.05
Average		65.38	18.92	3.94	11.76

Table 6 Summary of the best MVs distribution

Q_p	(**0**, **0**) (%)	(**X**, **0**) (%)	(**0**, **Y**) (%)	(**X**, **Y**) (%)
24	54.42	19.60	4.36	21.61
28	58.19	19.69	4.22	17.90
32	61.38	19.71	4.14	14.77
36	65.38	18.92	3.94	11.76
Average	59.84	19.48	4.17	16.51

From the comprehensive assessment on Table 6, we can see that (**0**, **0**) which is selected as the best MV accounts for the largest proportion, about 59.84 % on average. If it can be determined early, much more time will be saved. According to the principle that the best point is always center biased [17], we use an early termination strategy to stop early the block matching process. The early termination strategy performs an SDSP on (**0**, **0**) which represents the initial search point. If the initial search point is the best point which is with the minimum R-D cost among the five checked points, the following block matching search will be skipped. Otherwise, the following block matching search will be performed. The candidate MVs of SDSP are given by

$$\begin{aligned} MV_{SDSP\_candidate} &= \{(MV_x, MV_y)|(MV_x, MV_y) \\ &= (x \pm 1, y), (x, y \pm 1)\}, \end{aligned} \quad (3)$$

where (*x*, *y*) denotes the MV of initial search point or the MV of the best search point from previous step. An example of SDSP is shown in Fig. 3, where "circle" represents the initial search point or the best point which is obtained from the previous search step. "Triangle" represents the search points of SDSP.

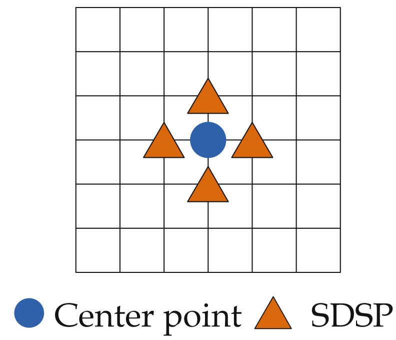


Fig. 3 Illustration of small diamond search pattern

From Table 6, we can also find that (**X**, **0**) and (**0**, **Y**) make up an significant proportion of the best MVs, with 19.48 and 4.17 %, respectively. On the other hand, (**X**, **0**) is about 4 times that of (**0**, **Y**), this reflects that the best MVs are asymmetrical distribution in natural video sequences. Based on the asymmetrical distribution between (**X**, **0**) and (**0**, **Y**), an ACSP and an SDSP are jointly used to find the best point. The search points in horizonal direction of ACSP is 4 times the number of that in vertical direction. The candidate MVs of ACSP are given by

$$\begin{aligned} MV_{ACSP\_candidate} &= \{(MV_x, MV_y)|(MV_x, MV_y) \\ &= (x \pm 2m, y), \quad m = 1, 2, \ldots S/2; \\ &\quad (x, y \pm 2n), \; n = 1, 2, \ldots S/8\}, \end{aligned} \quad (4)$$

where (*x*, *y*) represents the MV of initial search point. *S* is the search range. An example of ACSP is shown in Fig. 4, where "circle" represents the initial search point, "square" denotes the search points of ACSP. After getting the best point from ACSP, a SDSP is performed on the best point to detect whether it is the best MV with (**X**, **0**) or (**0**, **Y**). If the best point is at the center of SDSP, the following search will be skipped. Otherwise, locate the best MV from another half search points of the vertical direction of initial search point and (**X**, **Y**).

From Table 6, it is easy to see that the percentages of (**X**, **Y**) hold a relatively small value, with 16.51 % on average. In order to control the R-D performance, the best MVs with

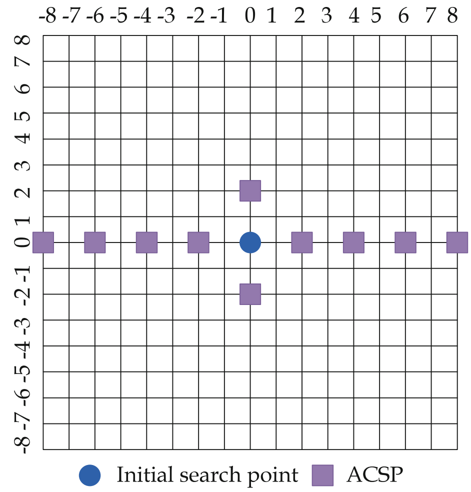


Fig. 4 Illustration of asymmetrical cross search pattern, search range = 8

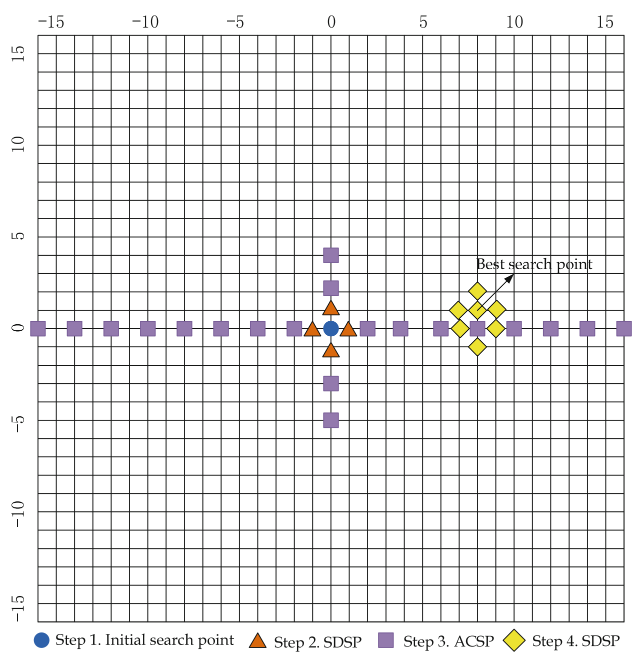


Fig. 5 Search procedure of the proposed fast ME algorithm, search range = 16

another half search points of the vertical direction of initial search point and (**X**, **Y**) should be located accurately. Due to this reason, we use a SDSP to detect the best MVs with them, and the SDSP is performed on the best point which is derived by the previous search step to determine the best MVs with (**X**, **0**) and (**0**, **Y**). If the search result locates at the center of SDSP, stop the following search and output the best MV. Otherwise, SDSP is recursively implemented until the best point locates at the center of SDSP.

Finally, the proposed fast ME algorithm is presented as Algorithm 1. An example of search procedure of the proposed algorithm is shown in Fig. 5, where "circle" is the initial search point. "Trianlge" represents the SDSP which is firstly used to terminate early the block matching search. "Square" is the search point of ACSP. "Diamond" denotes the recursive SDSP, which is secondly employed to terminate the block matching process, and locate the best matching block.

Algorithm 1 Proposed fast block matching algorithm based on MVs distribution

Step 1. Initial search point decision. MPM, INBPM and CBPM are applied to predict the initial search point. The predictive model with the minimum R-D cost among these three predictive models will be set as the initial search point.
Step 2. Perform a SDSP on the initial search point to early terminate the block matching process. If the best point has the minimum R-D cost is at the center of SDSP, go to Step 5; Else go to Step 3.
Step 3. Perform an ACSP on the initial search point. In this search procedure, we get a best point which is with the minimum R-D cost, and locates at $(\mathbf{X},\mathbf{0})$ or $(\mathbf{0},\mathbf{Y})$.
Step 4. The best point derived from Step 3 is set as the center to construct a SDSP. If the best point locates at the center position of SDSP, which means the best MV is with $(\mathbf{X},\mathbf{0})$ or $(\mathbf{0},\mathbf{Y})$, go to Step 5. Otherwise, SDSP is recursively implemented until the best point is at the center of SDSP, and locates the best MV from another half search points of the vertical direction of initial search point and $(\mathbf{X},\mathbf{Y})$.
Step 5. Return the best MV.

3 Experimental results

The proposed algorithm is implemented on H.264/AVC reference software JM 14.1 to evaluate its efficiency. The test conditions are listed in Table 1 and the search range is set to 16 and 32, respectively. The hardware platform is Intel E5800 3.16GHz CPU, 3.25GB RAM with Microsoft Windows 7 64-bit operating system. For QCIF, CIF, HD (720P) video sequences, 300, 200, 100 frames will be encoded, respectively.

3.1 Comparison of the hit rate

In order to evaluate the efficiency of the proposed algorithm, we compared the hit rate between the best MVs which are derived by the proposed algorithm and Xu's [18] improved UMHexagonS algorithm (we denote it as UM-HexagonS [18]). The best MVs which are obtained by FS are used as the benchmark MVs.

The results of three QCIF video sequences (*Akiyo*, *Bridge-Close* and *Claire*) as well as three CIF video sequences (*Hall*, *Mobile* and *Miss-America*) are presented in Fig. 6.

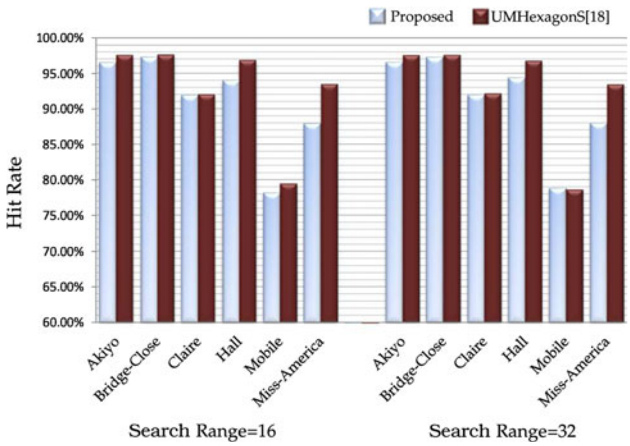


Fig. 6 Comparison of the hit rate between the proposed algorithm and UMHexagonS [18]

From Fig. 6, we can see that the proposed algorithm has a quite similar hit rate with UMHexagonS [18]. For video sequences with slow motion or simple context (such as *Akiyo* and *Bridge-close*, *Claire* and *Hall*), not only UMHexagonS [18] but also the proposed algorithm, the hit rate can reach up to 90 %. That is because (**0**, **0**) holds the largest proportion for these sequences. As for video sequences with fast motion activity (such as *Mobile*), there is a little decrease in hit rate, since the increased proportion of (**X**, **Y**). However, the decreased hit rate is still acceptable. Tables 7 and 8 show the R-D performance of some video sequences with fast motion activity (such as *Football*, *Bus*, *Mobile* and *Parkrun*), we can see that the R-D performance decrease is within an acceptable range. The comprehensive assessments indicate that, when the search range is equal to 16 and 32, the hit rate of the proposed algorithm is 91.23 and 91.03 % on average, respectively. The hit rate of UMHexagonS [18] is 92.89 and 92.72 %, when the search range is equal to 16 and 32, respectively. Compared to UMHexagonS [18], the hit rate of the proposed algorithm decreases about 0.2 %, 0.17 % on average, when the search range is equal to 16 and 32, respectively. These two values are quite acceptable and demonstrate that the proposed algorithm is effective in finding the best MVs.

3.2 Comparison of PSNR, bitrate and CPU time

We compare the performance of the proposed algorithm with recently fast ME algorithms, UMHexagonS [18] and Fast Directional Gradient Descent Search (FDGDS) [19] in terms of BDPSNR, BDBitrate [21], ME time and encoding time. The comparative results are tabulated in Tables 7 and 8. In these two tables, ΔMET and ΔEncT represent the CPU time reduction in ME process and encoding process, respectively. They are defined as

$$\Delta\mathrm{MET} = \frac{\mathrm{MET}_{\mathrm{proposed}} - \mathrm{MET}_{\mathrm{original}}}{\mathrm{MET}_{\mathrm{original}}} \times 100\,\%, \tag{5}$$

$$\Delta\mathrm{EncT} = \frac{\mathrm{EncT}_{\mathrm{proposed}} - \mathrm{EncT}_{\mathrm{original}}}{\mathrm{EncT}_{\mathrm{original}}} \times 100\,\%, \tag{6}$$

where $\mathrm{MET}_{\mathrm{proposed}}$ and $\mathrm{EncT}_{\mathrm{proposed}}$ denote the ME time and total encoding time of the the proposed algorithm. $\mathrm{MET}_{\mathrm{original}}$ and $\mathrm{EncT}_{\mathrm{original}}$ represent the ME time and total encoding time of UMHexagonS [18] and FDGDS, respectively.

Table 7 Summary of encoding results, search range = 16

Format	Sequence	Proposed vs UMHexagonS [18]				Proposed vs FDGDS [19]			
		BDPSNR (dB)	BDBitrate (%)	ΔMET (%)	ΔEncT (%)	BDPSNR (dB)	BDBitrate (%)	ΔMET (%)	ΔEncT (%)
QCIF	Coastguard	0.00	−0.07	−31.39	−16.57	0.02	−0.50	−7.56	−16.83
	Football	−0.14	2.23	−43.12	−26.83	0.25	−3.64	0.00	−3.43
	Hall	−0.04	0.84	−14.46	−6.96	0.03	−0.59	−3.30	−3.00
	Silent	−0.06	1.16	−23.20	−12.66	−0.02	0.41	−7.49	−7.17
	Bus	−0.07	1.31	−37.72	−23.03	0.90	−14.71	−3.62	−3.10
CIF	Container	0.00	0.06	−18.15	−9.97	0.01	−0.24	−6.37	−4.29
	Mobile	0.00	0.03	−39.65	−24.14	0.01	−0.32	−6.44	−5.10
	Paris	−0.05	1.01	−20.66	−8.91	0.00	−0.12	−7.57	−4.73
	Harbour	0.00	0.08	−29.59	−16.99	0.02	−0.43	−12.72	−13.18
	Parkrun	−0.01	0.11	−37.42	−21.09	0.02	−0.35	−28.49	−26.23
HD (720P)	Shields	−0.01	0.38	−31.22	−18.19	0.07	−2.60	−32.63	−29.04
	Spincalendar	0.01	−0.45	−24.18	13.40	0.00	−0.04	−20.35	−18.68
	Stockholm	−0.02	0.82	−28.04	−15.51	0.08	−4.68	−22.44	−19.26
Average		−0.03	0.58	−29.14	−14.42	0.11	−2.14	−12.23	−11.85

Table 8 Summary of encoding results, search range = 32

Format	Sequence	Proposed vs UMHexagonS [18]				Proposed vs FDGDS [19]			
		BDPSNR (dB)	BDBitrate (%)	ΔMET (%)	ΔEncT (%)	BDPSNR (dB)	BDBitrate (%)	ΔMET (%)	ΔEncT (%)
QCIF	Coastguard	0.00	−0.06	−58.84	−47.85	0.05	−1.34	−5.11	−0.72
	Football	−0.14	2.15	−50.58	−36.05	0.13	−2.00	−16.81	−12.62
	Hall	−0.01	0.20	−18.94	−10.71	0.00	−0.04	−12.53	−10.03
	Silent	−0.04	0.84	−40.60	−30.74	−0.01	0.24	−8.81	−4.54
	Bus	−0.08	1.55	−52.03	−39.18	0.85	−13.94	−7.17	−2.57
CIF	Container	−0.01	0.21	−32.42	−23.29	0.00	−0.04	−0.48	2.47
	Mobile	0.00	0.07	−46.50	−42.54	0.03	−0.64	−4.48	2.61
	Paris	−0.07	1.34	−28.56	−14.43	0.00	−0.15	−18.59	−14.68
	Harbour	0.00	0.04	−35.06	−21.76	0.00	−0.11	−20.66	−17.54
	Parkrun	0.00	−0.06	−46.79	−30.08	0.01	−0.30	−9.85	−2.51
HD (720P)	Shields	−0.01	0.51	−32.27	−19.44	0.06	−2.34	−24.63	−19.26
	Spincalendar	0.05	−0.92	−27.98	−16.20	0.00	0.25	−18.57	−14.28
	Stockholm	−0.02	1.14	−32.53	−19.77	0.06	−3.67	−6.47	−0.08
Average		−0.03	0.54	−38.70	−27.08	0.09	−1.85	−11.86	−7.21

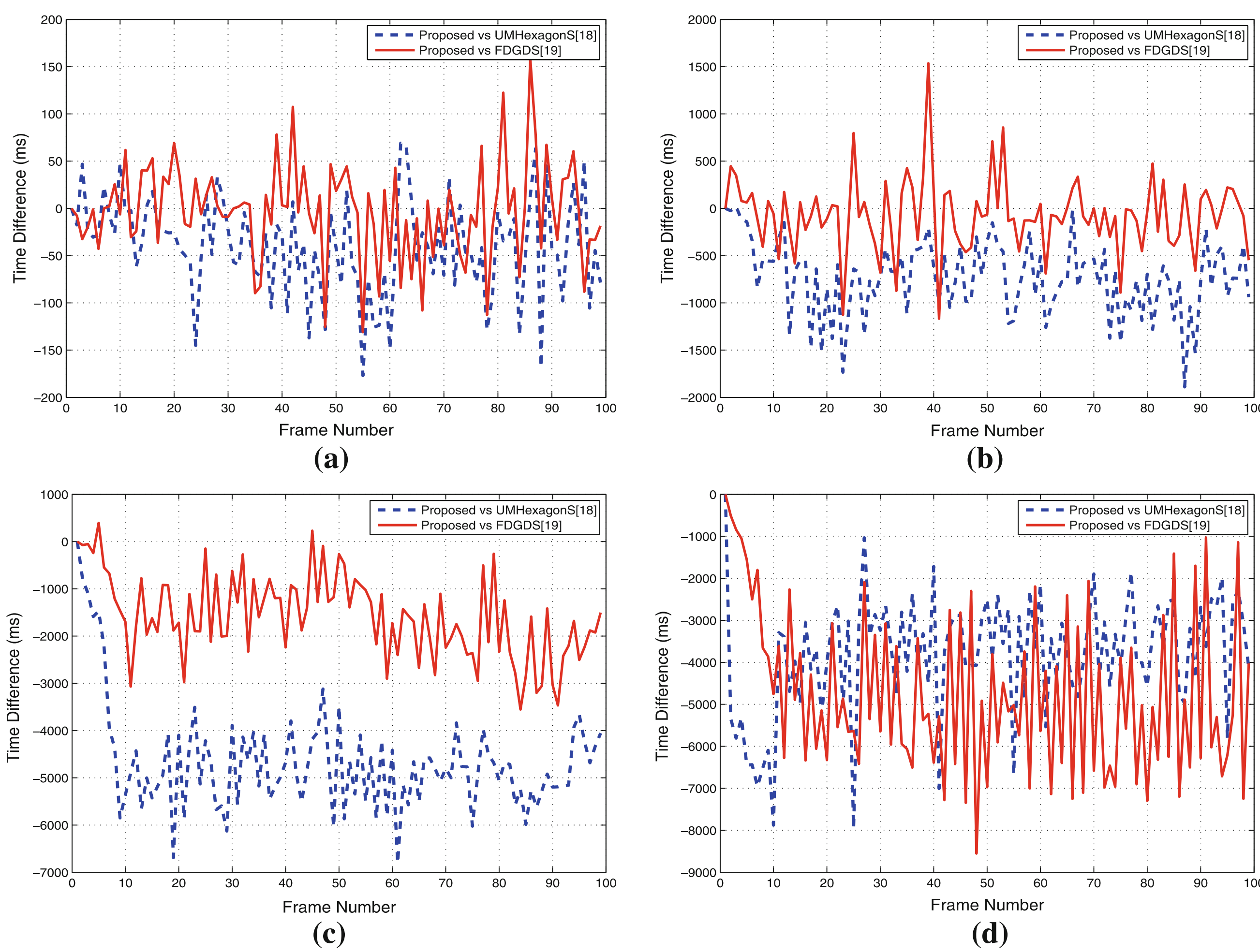


Fig. 7 Encoding performance difference in ME time, search range = 32. **a** *Hall*. **b** *Mobile*. **c** *Harbour*. **d** *Stockholm*

From Tables 7 and 8, we can see that our proposed fast ME algorithm can obtain significantly better results than UMHexagonS [18] and FDGDS in reducing the ME time and total encoding time. Moreover, our proposed algorithm obtains a better R-D performance than FDGDS. When search range equals to 16, compared to UMHexagonS [18], our algorithm can save the ME time and encoding time up to 29.14 and 14.42 % with 0.03 dB BDPSNR loss and 0.58 % BDBitrate increase. Compared to FDGDS, our algorithm can reduce the ME time and total encoding time up to 12.23 and 11.85 % with 0.11 dB BDPSNR increase and 2.14 % BDBitrate decrease. When search range equals to 32, our proposed algorithm can save the ME time and encoding time up to 38.70 and 27.08 %, BDPSNR loss 0.03 dB, BDBitrate increase 0.54 % as compared to UMHexagonS [18]. Compared to FDGDS, our proposed algorithm obtains a better R-D performance, 0.09 dB BDPSNR increase and 1.85 % BDBitrate decrease meanwhile the ME time and total encoding time reduce up to 11.86 and 7.21 % on average, respectively. For the video sequence (*Bus*), compared to FDGDS, our proposed algorithm achieves a quite better R-D performance, when search range equals to 16, BDPSNR increases about 0.90 dB and BDBitrate decreases −14.71 %. When search range equals to 32, the BDPSNR and BDBitrate change are 0.85 dB and −13.94 %, respectively. This is because the *Bus* sequence is with violent motion activities, FDGDS cannot locate the best search point accurately, and easily drop into local minimum.

Figure 7 shows the encoding performance difference of each frame in ME time of UMHexagonS [18], FDGDS and the proposed algorithm for four video sequences (*Hall* is with QCIF format and slow motion activity, *Mobile* is with CIF format and complex texture, *Harbour* and *Stockholm* are with HD format and medium motion activity) with search range equals to 32. From Fig. 7, we can see that our proposed algorithm consumes the minimum ME time for encoding each frame. Especially for HD sequences

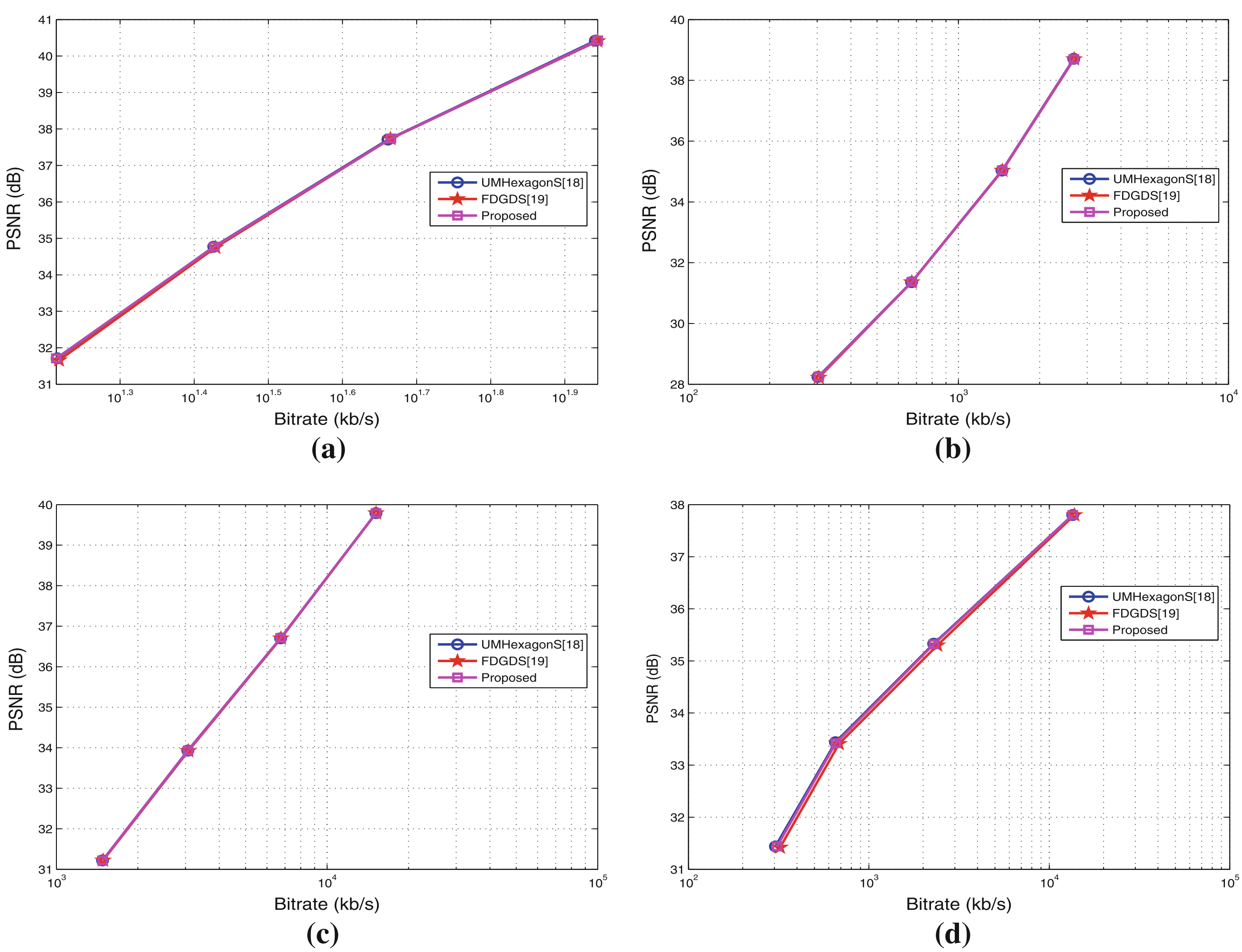


Fig. 8 R-D curves, search range = 32. **a** *Hall*. **b** *Mobile*. **c** *Harbour*. **d** *Stockholm*

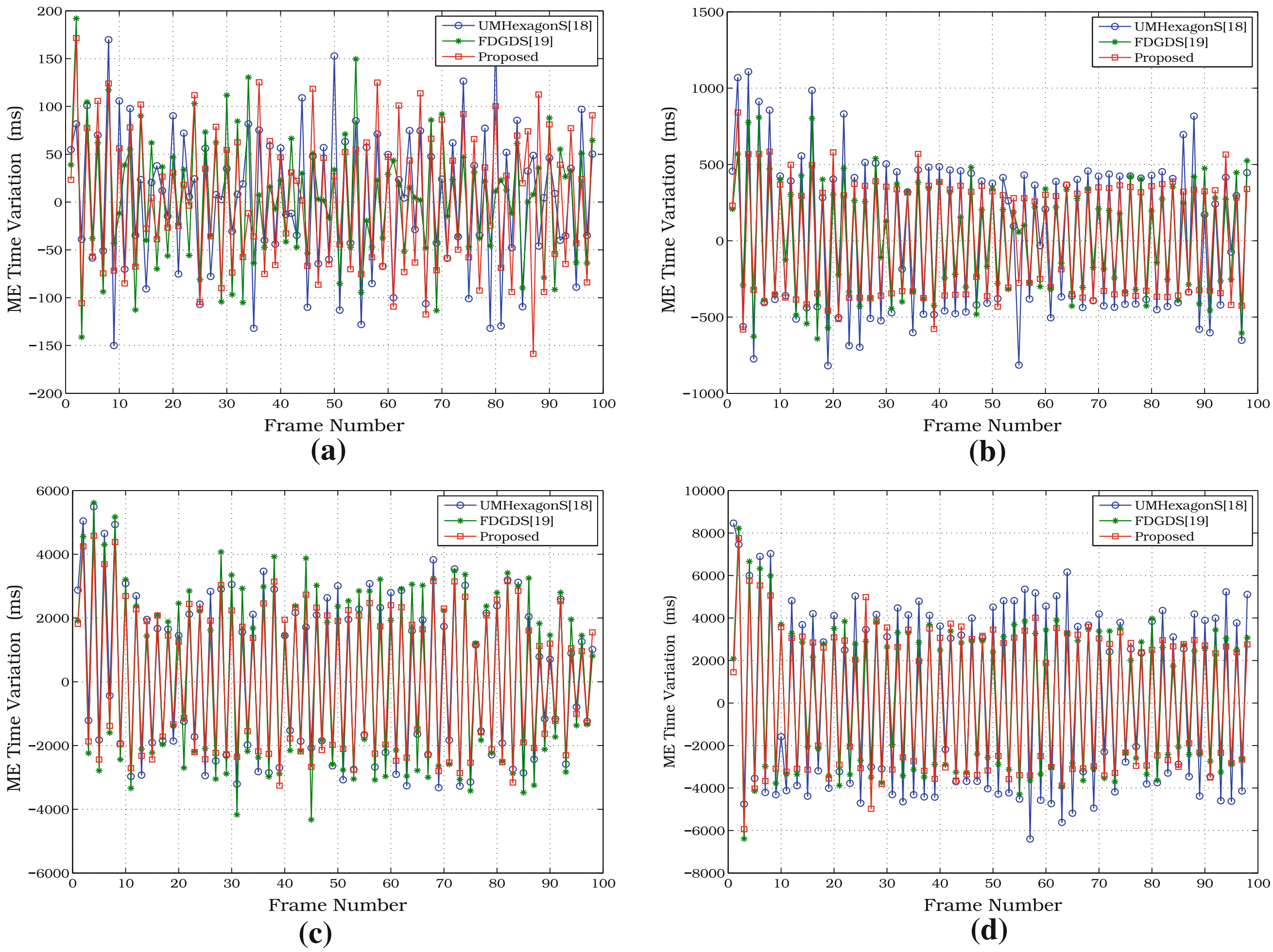


Fig. 9 Encoding performance in ME time variation, search range = 32. **a** *Hall*. **b** *Mobile*. **c** *Harbour*. **d** *Stockholm*

(*Harbour* and *Stockholm*), our proposed algorithm spends the least ME time when compared to UMHexagonS and FDGDS for encoding every frame. This demonstrates that our proposed algorithm can reduce the computational complexity of ME process efficiently. To demonstrate the R-D performance of the proposed algorithm, we give the R-D curves of *Hall*, *Mobile*, *Harbour*, *Stockholm* in Fig. 8. It can be observed that the proposed algorithm achieves a quite similar R-D performance as the compared algorithms. For *Stockholm*, the proposed algorithm achieves a better R-D performance than FDGDS. From Figs. 7 and 8 and Tables 7 and 8, we can conclude that our proposed algorithm can work well in not only video sequences with slow motion or simple context (such as *Hall*, *Silent*, *Container*) but also the sequences with fast motion and complex context (such as *Coastguard*, *Mobile* and *Parkrun*). Especially, for HD video sequences, the proposed algorithm can achieve a relatively better coding performance.

In order to demonstrate that our proposed algorithm is suitable for real-time application, we give a comparison on ME time variation between UMHexagonS, FDGDS and our proposed algorithm. The results of four video sequences with different motion activities and texture (*Hall*, *Mobile*, *Harbour* and *Stockholm*) are shown in Fig. 9, and the ME time variation is measured by the difference between two consecutive frames. From Fig. 9, it can be observed that the variation scope of our proposed algorithm is almost the smallest when compared to UMHexagonS and FDGDS. For the video sequence (*Hall*), the variation scope of these three algorithms are not stable. For the other three video sequences (*Mobile*, *Harbour* and *Stockholm*), our proposed algorithm is with the smallest variation scope for most frames. Generally, we can see that our proposed algorithm consumes relatively less ME time for encoding each frame and the ME time variation is relatively more stable. Hence, compared to UMHexagonS and FDGDS, our proposed algorithm is more suitable for real-time application. However, for video sequence *Hall*, the ME time variation is changed dramatically in some frames.

The ME time variation depends on the video content, prediction structure and ME algorithm. Since video content and prediction structure are fixed, we can address the fluctuation of ME time variation from the algorithm aspect. Firstly, total ME time can be reduced by jointly optimizing the MD, MRFS and searching best point process. Moreover, according to the best MVs distribution in the previous frame, the search range of ME in the current frame is possible to be adaptively reduced; hence much more ME time could be saved. Secondly, to dynamically limit the ME time of each frame, the given ME time (T) can be allocated to each MB, then define several thresholds T_n. Initially, the fast ME is performed by considering R-D performance with high priority. If the best point is not achieved within time threshold, T_n, faster ME strategies or ME early termination can be adopted to guarantee that the ME time will not exceed the given time T. Take our proposed algorithm as an example, in step 2, the SDSP can be simplified based on the MB movement direction, if the MB deflects to horizonal direction, only two horizonal points of SDSP are performed. Otherwise, two vertical points are searched. At last, the time can be reduced by half; in step 3, the time can also be saved by adjusting the interval of ACSP from 2 to 4 or larger. More sophisticated control algorithms are possible and ideas can be adopted from rate control and automatic control theory. Generally, while designing the real-time application system, we can meet the real-time requirements at the cost of hardware resources by using parallelization, or using high frequency devices (CPU/memory/bus), etc. Here, in the algorithm aspect of the encoder, we can adopt some control algorithms, early termination and new ME search strategies (giving time reduction with high priority) to dynamically meet ME time limitation. However, it may at the cost of R-D performance.

4 Conclusion

In this paper, an efficient fast ME algorithm is proposed. Firstly, based on the correlations between spatial and temporal blocks, three predictive models are jointly employed to determine the initial search point. After getting the initial search point, according to distribution of the best MVs in natural video sequences, an SDSP and an ACSP are used to locate the best matching block. The experimental results demonstrate that the proposed algorithm yields a quite promising performance in terms of R-D performance and computational complexity. The experimental results also show that the proposed algorithm works well in different formats of video sequences with various motion activities, and is more suitable for real-time application.

Acknowledgments This work is partly supported by the City University of Hong Kong Strategic Grant 7002729.

References

1. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 560–576 (2003)
2. Advanced Video Coding for Generic Audiovisual Services, ISO/IEC 14496–10: 2005(E) ITU-T Rec. H.264(E) (2005)
3. Ren, J., Kehtarnavaz, N., Budagavi, M.: A fast feature-assisted adaptive early termination approach for multiple reference frames motion estimation in H.264. J. Real Time Image Process. **3**(1–2), 77–88 (2008)
4. Huang, Y.W., Hsieh, B.Y., Chien, S.Y., Ma, S.Y., Chen L.G.: Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC, IEEE Trans. Circuits Syst. Video Technol. **16**(4), 507–522 (2006)
5. Koga, T., Iinuma, K., Hirano, A., Iijima, Y., Lshiguro T.: Motion-compensated interframe coding for video conferencing. In: Proceeding of the National Telecommunications Conference, New Orleans, G5.3.1–G5.3.5 (1981)
6. Po, L.M., Ma, W.C.: A novel four-step search algorithm for block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **6**(3), 313–317 (1996)
7. Liu, L.K., Feig, E.: A block-based gradient descent search algorithm for block motion estimation in video coding. IEEE Trans. Circuits Syst. Video Technol. **6**(4), 419–422 (1996)
8. Zhu, S., Ma, K.K.: A new diamond search algorithm for fast block matching motion estimation. IEEE Trans. Image Process. **9**(2), 287–290 (2000)
9. Ranjit, S.S.S., Sim, K.S., Besar, B., Md Salim, S.I., Subramaniam, S.K.: Estimation of motion vector parameter using hexagon-diamond search algorithm. J. Real Time Image Process. **6**(4), 225–234 (2011)
10. Zhu, C., Lin, X., Chau, L.: Hexagon-based search pattern for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **12**(5), 349–355 (2002)
11. Po, L.M., Ting, C.W., Wong, K.M., Ng, K.H.: Novel point-oriented inner searches for fast block motion estimation. IEEE Trans. Multimed. **9**(1), 9–15 (2007)
12. Chen, Z., Zhou, P., He, Y.: Fast interger pel and fractional pel motion estimation for JVT, Document JVT-F017, 6th meeting. Awaji Island, JP (2002)
13. Chen, Z., Xu, J., He, Y., Zheng, J.: Fast integer-pel and fractional-pel motion estimation for H.264/AVC. J. Visual Commun. Image Represent. **17**(2), 264–290 (2006)
14. Tourapis, A.M., Au, O.C., Liou, M.L.: Highly efficient predictive zonal algorithms for fast block-matching motion estimation. IEEE Trans. Circuits Syst. Video Technol. **12**(10), 934–947 (2002)
15. Lim, K.: Text description of joint model reference encoding methods and decoding concealment methods, Document JVT-O079, 15th meeting. Busan, Korea (2005)
16. JVT H.264/AVC reference software version JM14.1. http://iphome.hhi.de/suehring/tml/download/. Accessed 6 Jan 2011
17. Li, R., Zeng, B., Lio, M.L.: A new three-step search algorithm for block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **4**(4), 438–443 (1994)

18. Xu, X., He, Y.: Improvements on fast motion estimation strategy for H.264/AVC. IEEE Trans. Circuits Syst. Video Technol. **18**(3), 285–293 (2008)
19. Po, L.M., Ng, K.H., Cheung, K.W., Wong, K.M., Uddin, Y.M., Ting, C.W.: Novel directional gradient descent searches for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol. **19**(8), 1189–1195 (2009)
20. Pan, Z., Kwong, S., Zhang, Y.: A multiple hexagon search algorithm for motion and disparity estimation in multiview video coding. The Era of Interactive Media, pp. 94–103. Springer, Berlin (2012)
21. Bjontegaard, G.: Calculation of average PSNR differences between RD-curves, Document VCEG-M33, VCEG 13th meeting. Austin, Texas, USA (2001)

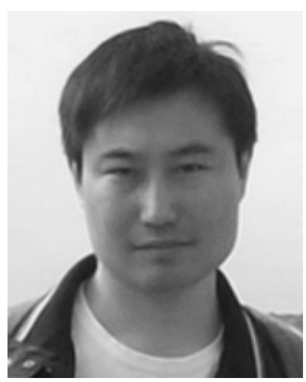
Author Biographies

Zhaoqing Pan received his B.S. degree in Computer Science and Technology with honors from Yancheng Normal University, Yancheng, China, in June 2009. Currently, he is a Ph.D. candidate in the Department of Computer Science at the City University of Hong Kong, Kowloon, Hong Kong. His research interests include motion estimation and mode decision in video coding.

Sam Kwong (M'93-SM'04) received the B.S. and M.S. degrees in electrical engineering from the State University of New York at Buffalo in 1983, the University of Waterloo, Waterloo, ON, Canada, in 1985, and the Ph.D. degree from the University of Hagen, Germany, in 1996. From 1985 to 1987, he was a Diagnostic Engineer with Control Data Canada. He joined Bell Northern Research Canada as a Member of Scientific Staff. In 1990, he became a Lecturer in the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor in the Department of Computer Science. His research interests are video and image coding and evolutionary algorithms.

Long Xu received his M.S. degree in Applied Mathematics from Xidian University, China, in 2002, and his Ph.D. degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). He was a Postdoc researcher at the Department of Computer Science, City University of Hong Kong from August 2009 to July 2011. And now, he is working for the Department of Electronic Engineering of Chinese University of Hong Kong as a Postdoc researcher. His research interests include image/video coding, wavelet-based image/video coding and computer vision.

Yun Zhang received the B.S. and M.S. degrees in electrical engineering from Ningbo University, Ningbo, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2010. From 2009 to 2011, he was a Postdoc Researcher with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. In 2010, he joined in the Shenzhen Institutes of Advanced Technology (SIAT), CAS, as an Assistant Professor. His research interests are multiview video coding, video object segmentation and content-based video processing.



Tiesong Zhao (M'12) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2006 and the Ph.D. degree in computer science from the City University of Hong Kong, Kowloon, Hong Kong, in 2011. He is currently a Research Associate in the Department of Computer Science, City University of Hong Kong. His research interests include signal processing, digital video coding, and fast algorithms.