

# On the Design of Pattern-based Block Motion Estimation Algorithms

Jang-Jer Tsai, *Member, IEEE*, and Hsueh-Ming Hang, *Fellow, IEEE*

**Abstract**—Pattern-based block motion estimation (PBME) is a critical element in the contemporary video coding system because it typically dominates the coding efficiency and the computing power. Therefore, many proposals have been suggested to reduce its computational complexity, but most of them are devised based on experimental data or heuristic ideas. In this letter, we look into every component of a typical PBME algorithm and fine tune the major components systematically to achieve the optimal or nearly optimal results. Our methodology is developed based on our proposed analytical model together with statistical tools. First, we use the analytic model to analyze and design effective genetic-algorithm-based search patterns. Moreover, we propose an adaptive switching strategy that dynamically switches between two search patterns. Second, we extend our PBME model to evaluate the efficiency of starting (initial search) points. A near optimal set of starting points is progressively identified. Last, we study the early termination threshold technique and suggest a metric in selecting an effective threshold. An accurate threshold mechanism is thus constructed. Combining all these techniques, we develop a PBME algorithm that outperforms most popular algorithms.

**Index Terms**—Early termination, genetic pattern searches, modeling, motion estimation, starting points.

## I. INTRODUCTION

MODERN video compression systems convert the huge digitized video data into a small-size sophisticated bit-stream by using the well-known block based hybrid coding (BHC) structure [1]. In general, a BHC video system comprises two major modules: intra frame coding and inter frame coding. Block based motion estimation (BME) algorithm plays the key role in the inter-frame coding. Yet, BME is computational intensive; thus, a myriad of fast BME have been proposed.

The most popular class of the BME algorithm is the pattern-based block motion estimation (PBME), which is typically a multi-step process. Often, three sets of tools are included: (1) search patterns [3]–[10], (2) starting points [3], [4], [18], [19],

and (3) early termination thresholds [3], [4], [18], [20]. Despite the prosperity of PBME, most PBME algorithms are devised based on heuristics and speculation on experimental data. In this letter, we use the analytical model in [2] to scrutinize the underneath mechanism in each tool. Then, by adopting a step-by-step systematic approach, we improve each tool. At the end, we combine these tools together to form a very effective PBME algorithm.

The rest of this letter is organized as follows. Section II reviews our previously proposed analytical model [2] for PBME algorithms. In Section III, we propose two sets of the genetic-algorithm-based search patterns for different types of moving image sequences. Then, we design a pattern switching strategy, which dynamically changes search patterns based on the real-time video statistics. Section IV examines the impact of starting (initial) point set and suggests a starting point set that produces outstanding search results. Section V suggests a threshold predictor that can be used in the early termination algorithm. Combining all these techniques together, Section VI presents a complete PBME algorithm and its performance.

## II. MODELING OF PBME ALGORITHMS

$$ASP = C_1 \times \sum_{x,y \in A} S_{FS}(x,y) \times WF_{SA}(x,y) + C_2 \quad (1)$$

$$S_{FS}(x,y) = \frac{\frac{1}{|x|^{5/3} + \zeta_x} \frac{1}{|y|^{5/3} + \zeta_y}}{\sum_{x',y' \in A} \frac{1}{|x'|^{5/3} + \zeta_x} \frac{1}{|y'|^{5/3} + \zeta_y}} \quad (2)$$

$$PMV = \text{median}(MV^L, MV^U, MV^{UR}) \quad (3)$$

where  $MV^L$ ,  $MV^U$ , and  $MV^{UR}$  are the motion vectors of the left, up, and up-right block neighbors to current block, as illustrated by Fig. 9.

In [2], we propose a mathematical model [expressed by (1)] that can predict the average number of search points (ASP) produced by a PBME scheme. This model consists of two components: a probability distribution function  $S_{FS}(x,y)$  of MVs [approximated by (2)], and the minimal number of search points needed to identify an MV located at  $(x,y)$ ,  $WF_{SA}(x,y)$  (called weighting function) which is search algorithm (SA) dependent. In (1),  $(x,y)$  are the relative coordinates of which the origin is the predicted motion vector [PMV, defined by (3)]. The parameters  $(C_1, C_2)$  are obtained empirically by training methods. Note that  $C_1$  is always positive because ASP

Manuscript received May 13, 2008; revised July 3, 2008 and November 25, 2008. First version published July 7, 2009; current version published January 7, 2010. This paper was recommended by Associate Editor O. C. Au.

J.-J. Tsai is with the Department of Electronics Engineering, National Chiao-Tung University, Hsin-Chu 300, Taiwan (e-mail: jjtsai@aa.nctu.edu.tw).

H.-M. Hang is with the Department of Electronics Engineering, National Chiao-Tung University, Hsin-Chu 300, Taiwan, and with the Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 106, Taiwan (e-mail: hmhang@mail.nctu.edu.tw).

Color versions of one or more of the figures in this letter are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2009.2026805

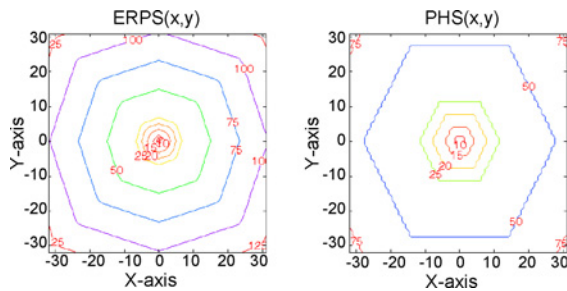


Fig. 1. Contour plots of WF for ERPS and PHS.

 TABLE I  
 THE TEST SEQUENCES AND THEIR SETTINGS

Abbreviation	Sequence	Bit Rate (kb/s)	Frame Rate (frames/s)	Number of Frames	PSNR
CT256	<i>Container</i>	256	7.5	300	39.56
CT40	<i>Container</i>	40	7.5	300	32.04
HL40	<i>Hall</i>	40	7.5	300	33.55
MD96	<i>Mother and Daughter</i>	96	10	300	39.80
CG112	<i>Coastguard</i>	112	30	300	29.08
FM512	<i>Foreman</i>	512	30	300	34.06
FM1024	<i>Foreman</i>	1024	30	300	36.56
FB1024	<i>Football</i>	1024	30	90	35.28
FG768	<i>Flower Garden</i>	768	30	250	26.33
ST1024	<i>Steven</i>	1024	30	300	29.48

and the sum of products of  $S_{FS}(x, y)$  and  $WF_{SA}(x, y)$  are always positively correlated.

Equation (2) is derived based on the motion vectors acquired from image data. In (2),  $(x, y)$  and  $(x', y')$  are relative coordinates with respect to PMV, and  $A$  is the search area. The parameters  $(\zeta_x, \zeta_y)$  are obtained by numerical methods so that the variances of  $S_{FS}(x, y)$  match those of the MVs acquired by full search (FS) applied to a specific sequence.

The weighting function,  $WF_{SA}(x, y)$ , is the minimal number of search points produced by a specific PBME algorithm when the argument  $(x, y)$  is the target MV. The weighting function is typically obtained by analyzing the search procedure. Fig. 1 shows the contour plot of the weighting functions of two popular pattern search algorithms, point-oriented hexagonal search (PHS) [7] and easy rhombus pattern search (ERPS). The ERPS adopted here is the adaptive rood pattern search in [4] but using PMV as the sole starting point. The value marked on a contour represents the minimal search points required for a search algorithm to move from the origin to a point on the contour. The weighting function is a discrete function, and the data points exist only on the integer coordinates. For the ease of visualization, the data points are interpolated to form continuous contours. Details of this model are referred to [2].

Table I shows the test sequences in our experiments, coded at typical bit rates (to produce acceptable PSNR quality). They are coded by a MPEG-4 SP@L3 encoder. All the sequences are in the CIF ( $352 \times 288$ ) format. Only the first frame is coded as the I frame, and all the remaining frames are coded as the P frames. The motion vector search range is chosen to

be 16, the initial quantization step size is set to 15, and the block size is  $16 \times 16$ . The quantization step is adjusted by a rate controller to achieve the desired bit rate. The coded video peak signal-to-noise ratio (PSNR) ranges roughly from 26 dB (poor yet acceptable) to 40 dB (visually the same as original). The frame skip and the block skip (macroblock not coded) modes are not in use.

### III. ADAPTIVE GENETIC PATTERN SEARCH ALGORITHMS

#### A. Genetic Search Patterns

A preferred pattern search should have the following desirable properties: 1) it consumes less computing power, 2) it does not degrade the video quality, and 3) it costs fewer bits in coding the MV vectors.

In [2] and [10], after analyzing the weighting functions of several popular search algorithms (Fig. 1), we find that ERPS has the smallest ASP values for the MVs near the origin (PMV) and PHS has the smallest ASP for the points away from the origin. These observations are consistent with the well-known facts that PHS moves faster than many other algorithms, thus can quickly reach the distant locations, and ERPS examines fewer points when the target MV is close to the origin. Ideally, a good PBME algorithm should have small weighting function values for all locations in the search area, particularly for the high probability target MVs.

A search algorithm degrades the video quality when it is trapped into a local optimum point. To reduce such cases, a search algorithm shall check all neighboring points of the target when it decides to terminate the search process. The dilemma is that the increased checking points also increase computation. To achieve a balance between speed and quality, a PBME algorithm shall carefully select the number and the locations of check points at the termination step.

A search algorithm should make good use of the uneven MV distribution to reduce the entropy coding bits. For example, if the (best) MVs cluster around a predictable location, it takes fewer bits in encoding MVs and less computing power in finding MVs. Because the probability density function of typical MVs peaks at around the PMV, a PBME algorithm with small weighting function near the starting point (PMV) would consume less computing power and fewer coding bits on the average. For convenience, therefore, our PBME model is centered at PMV.

Based on the above design considerations, we adopt the genetic algorithms [22] to modify the traditional PBME algorithms. The simplest genetic algorithm contains only a mutation-and-competition loop. When a survivor (parent) produces a mutant (a child), the survivor competes with its own mutant to decide the next survivor (next-stage parent). The process stops when the survivor beats all its mutants. In contrast, the traditional PBME algorithms check all points in the search pattern and move the center (origin) to the winner until the central point beats all the other points in the search pattern.

A traditional PBME algorithm typically consists of two search patterns, the large search pattern and the small search pattern. The large search pattern is used for the coarse (regular) search and the small search pattern is used for the fine

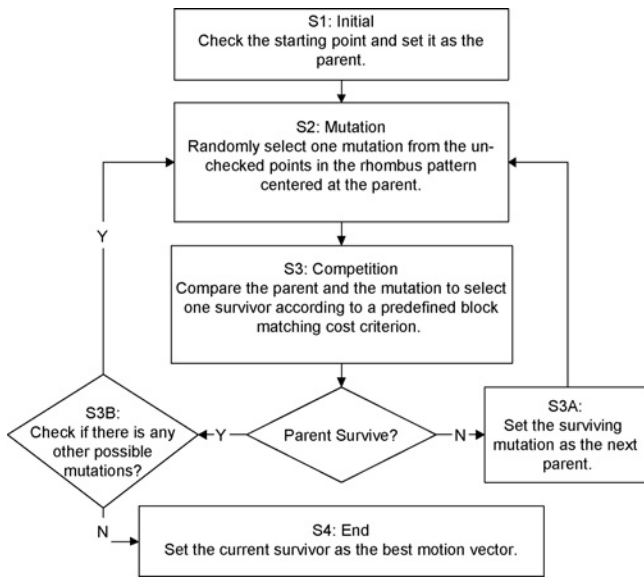


Fig. 2. Flowchart of GRPS.

(terminating) search. In converting a traditional search algorithm into a genetic one, we blend the genetic algorithm into the coarse search stage. The central point (which is the winner of the previous search step) in the search pattern is the parent in the genetic search and all the other points are the child candidate set. Instead of calculating the block matching cost of all the child candidates and deciding the best MV, we randomly select a point (a mutant) from the child candidate set, calculate its block-matching cost, compare its cost with the parent's cost (*competition*), and decide the survivor (next *parent*). This process continues until all the points in the current child set are examined. If the parent beats all its children, it is then declared to be the winner. In addition, a typical terminating search checks all the points in the small search pattern to avoid trapping into the local minimum. But recent studies [7] suggest that it is often sufficient to check only the candidate points near the smallest error points in the large pattern.

Because of the computational advantage of the genetic algorithm, we convert ERPS into GRPS (genetic-based ERPS) and PHS into GPHS (genetic-based PHS), respectively.

The flowchart of GRPS is shown in Fig. 2, and its associated search pattern is shown in Fig. 3. In Step 2 (S2), it randomly checks one point (black, for example) among all search points in Fig. 3(a). The condition of Step 3B (S3B) is whether all the (black) points in Fig. 3(b) have been checked.

The flowchart of GPHS is shown in Fig. 4 and its associated search patterns are shown in Fig. 5. Steps 2 and 3 (S2 and S3) are similar to those of GRPS but with a different large search pattern. In Step 4 (S4, refinement), as suggested in [7], we first calculate the cost function, so-called normalized group distortion (NGD) defined by (4) in [7], for all the grey points in Fig. 5(b). Then, we select the smallest NGD point from points a to f, and the smaller NGD point from points g and h. These two points constitute the small search pattern. Herein, the NGD of points a to h is calculated using the SADs in the Groups A–H in Fig. 5(c) and (d), respectively.

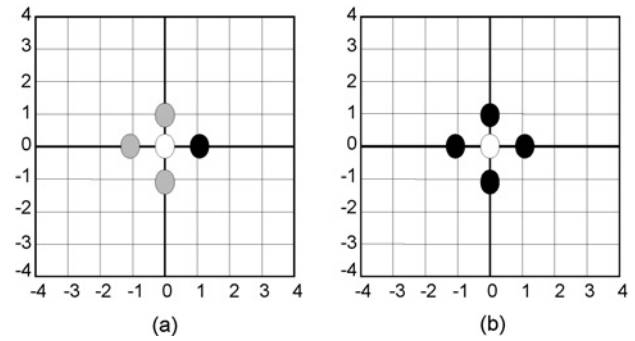


Fig. 3. Search patterns for GRPS.

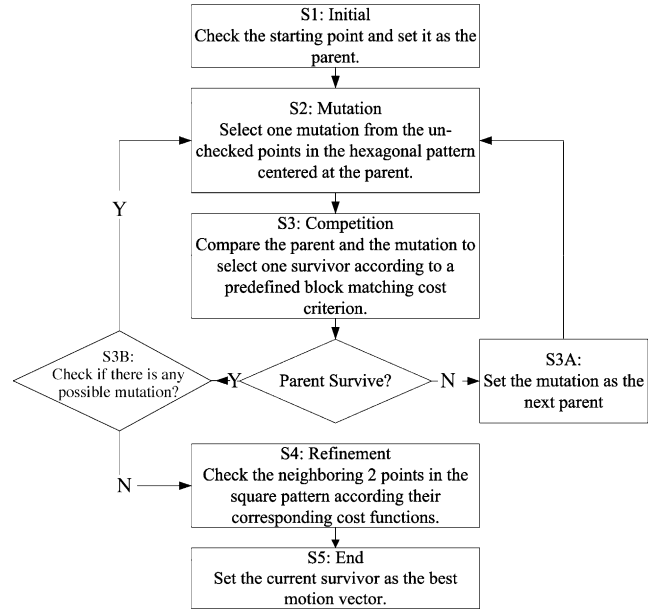


Fig. 4. Flowchart of GPHS.

Fig. 6 shows the WF of GRPS and GPHS. Compared with the WF of the nongenetic PBME algorithms in Fig. 1, GRPS has the smallest values around the center but GPHS has the smallest values at far-away locations. Experimental results verify our above observations on WF.

Moreover, the computational overhead of the genetic-based algorithms is very small because we do not use the entire conventional genetic algorithm. Other than a few additional comparisons of the matching errors, the only computational overhead is the random selection of a mutant from the child set and this process can be implemented by a simple pseudo number generator.

### B. Adaptive Pattern Switching Strategy

Because the contents of video sequences vary drastically, one single search pattern may not produce the best result in terms of speed and PSNR. Thus, the adaptive pattern-switching search algorithms were proposed [11]–[17]. These algorithms are empirically constructed and the switching criterion is often based on block (feature) classification. Few papers have clear and strong evidence as why certain block features can be used as the switching criterion. Also, there are few discussions on

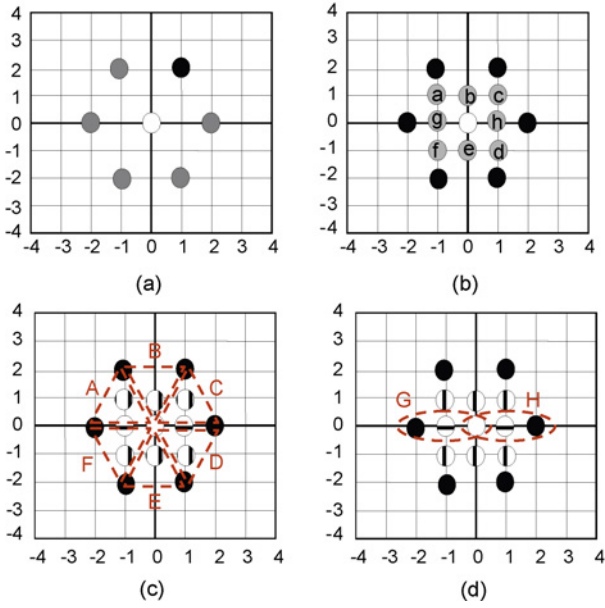


Fig. 5. Search patterns of GPHS.

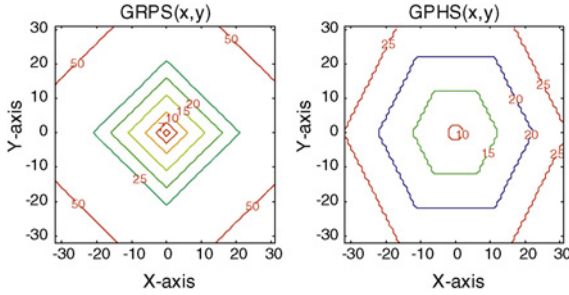


Fig. 6. Contour plots of the weighting function for GRPS and GPHS.

how to optimally choose the search pattern set. Therefore, we like to explore these issues based on our previous study [17].

We look for an adequate index that can be used to decide the right instant to switch between two search patterns. The target is lowering the computational complexity. That is, if Search Algorithm 1 (SA1) is in use, would the average search points be fewer than that produced by using Search Algorithm 2 (SA2)? Based on our average search point (ASP) model, (1), the difference in ASP is expressed by (4). Note that  $WF_{SA1}$  and  $WF_{SA2}$  depend only on search algorithms (SA). But because  $S_{FS}$  is a function of the MV variance,  $D_{ASP}$  is thus picture-dependent. The parameter  $C_1$  is fixed for a video sequence. Dividing  $D_{ASP}$  by  $C_1$ , we obtain the *switching index* ( $I_{ASP}$ ) defined by (5).

$$D_{ASP} = C_1 \times \sum_{x,y \in A} S_{FS}(x,y) \times (WF_{SA1}(x,y) - WF_{SA2}(x,y)) \quad (4)$$

$$I_{ASP} = D_{ASP}/C_1. \quad (5)$$

GRPS and GPHS are chosen as the basic search patterns owing to their short-range and long-range search performance. Then, the  $I_{ASP}$  between GRPS and GPHS, drawn against

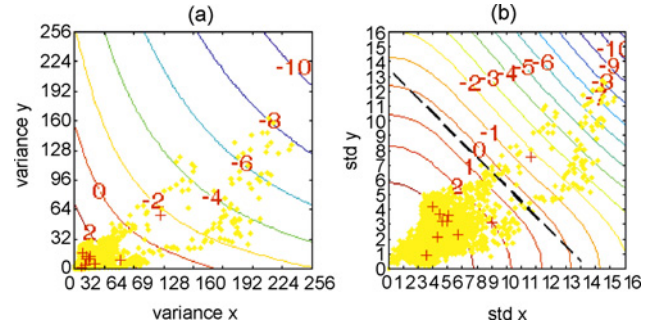


Fig. 7. IASP between GRPS and GPHS w.r.t. MV variance or MV standard deviation.

two variables, MV variance or MV standard deviation, are the solid contours in Fig. 7. In Fig. 7(a), the  $x$ -axis is the variance of the MV horizontal component and the  $y$ -axis is the variance of the MV vertical component. In Fig. 7(b), the axes are the MV standard deviations along the horizontal direction and the vertical direction, separately. When  $I_{ASP} > 0$ , GRPS outperforms GPHS in terms of ASP, and when  $I_{ASP} < 0$ , GPHS is better. Therefore, the switching criterion is the boundary that  $I_{ASP}$  equals zero. For the GRPS and GPHS pair, the threshold,  $I_{ASP} = 0$ , is approximated by a straight black dashed line (6) in Fig. 7(b). The red plus marks denote the MV variances or standard deviations of the test video sequences and the yellow dots denote the MV variances or standard deviations of the frames in the test video sequences. Because the  $I_{ASP}$  of most test sequences and frames are larger than 0, GRPS is chosen more often. Only in the extreme cases, GPHS stands out.

$$U \cdot STD_x + V \cdot STD_y = W. \quad (6)$$

In the real-time applications, the MV standard deviations of the current frame are not available before its MVs are all calculated. Fortunately, the motion characteristics in an image sequence typically change gradually [3]; therefore, the MV standard deviations in the neighboring spatial or temporal areas are generally similar. After testing a few MV standard deviation predictors, we found that the MV standard deviations of the previous frame are good predictors to its values in the current frame.

Furthermore, the MV characteristics may vary in different parts of a frame. Hence, we can switch the search pattern for each block. Because the MV characteristics in the nearby spatial/temporal area tend to be similar, the standard deviations of motion vector  $MV^L$ ,  $MV^U$ , and  $MV^P$  in Fig. 9 are used for the block level pattern switching criteria. The so-called double level pattern switching strategy for AGPS (abbr. DL AGPS) is thus proposed and its flowchart is shown in Fig. 8. If the previous frame has small MV standard deviations ( $MV\_STD_{frame}^x$  and  $MV\_STD_{frame}^y$ ), we incline toward using GRPS as the search pattern with the exception that the MV standard deviations derived from the nearby blocks ( $MV\_STD_{block}^x$  and  $MV\_STD_{block}^y$ ) are very large. On the other hand, if the previous frame has large MV standard deviations, GPHS is often chosen unless the MV standard deviations derived from

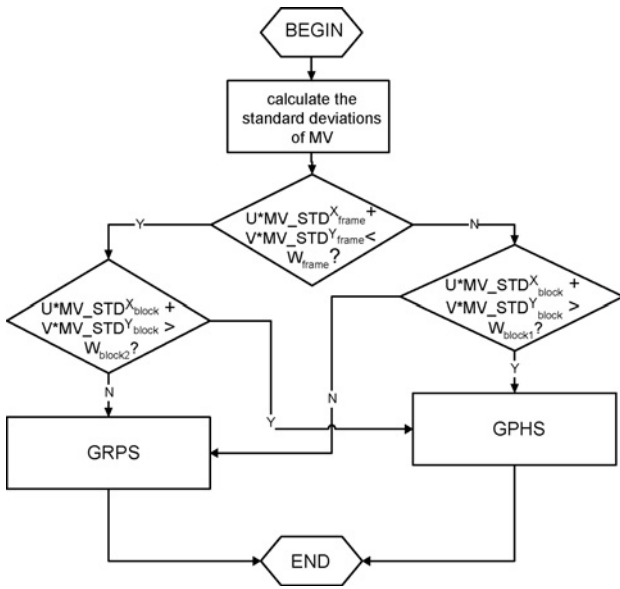


Fig. 8. Flowchart of the double level adaptive genetic pattern search (DL AGPS).

the neighboring blocks are very small. The parameter values of  $U$ ,  $V$ ,  $W_{\text{frame}}$ ,  $W_{\text{block1}}$ , and  $W_{\text{block2}}$  are derived from data by using the numerical method. In our experiments,  $U = 1$ ,  $V = 1$ ,  $W_{\text{frame}} = 12$ ,  $W_{\text{block1}} = 8$ , and  $W_{\text{block2}} = 16$ .

The computational overhead of the proposed adaptive pattern selection strategy is very small. At the frame level, the frame MV variance/standard deviation is calculated once per frame. At the block level, we only use the upper, the left, and the collocated block motion vectors to calculate the MV variance/standard deviation. In computer simulation, the run time profiling shows that the overhead of the proposed adaptive strategy consumes only about 2% of the total computation used by the ME module.

#### IV. STARTING POINT SELECTION

The impact of starting points or initial points on fast search algorithms has been studied by many researchers [3], [4], [18], [19]. Typically, the starting point is predicted by using a combination of the MVs of several neighboring blocks. The most probable MV estimated by this type of MV predictor is used as the starting point for PBME algorithms. Although many MV predictors have been suggested, they are derived mostly based on experimental data. Here, we design a criterion that evaluates the effectiveness of MV predictors and propose a systematical approach that constructs the optimal starting point set (SPS).

We again use the proposed PBME model (first method in [2]) in solving the starting point selection problem. Because the MV field acquired by FS is fixed for a given video sequence, a different starting point only does a translational shift on the motion vector distribution. Given two starting points, SP1 and SP2, their difference in ASP ( $E_{\text{ASP}}$ ) can be represented by (7). Let SP2 be a fixed starting point for comparison purpose; (7) thus becomes (8), in which  $\eta$  is a constant. Rearrange (8), we obtain  $G_{\text{ASP}}$  defined by (9), which

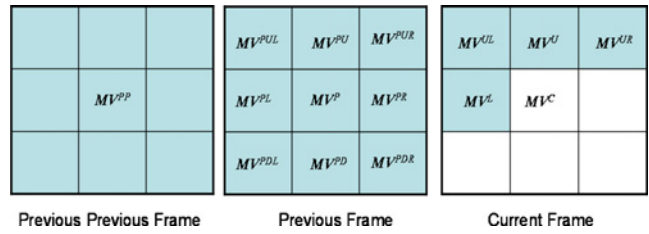


Fig. 9. Motion vector predictor candidates in the current frame, the previous frame, and the frame before previous frame.

is proportional to the ASP using SP1. Thus,  $G_{\text{ASP}}$  is used as the performance assessment criterion for starting point evaluation.

$$E_{\text{ASP}} = C_1 \times \sum_{x,y \in A} ((S_{\text{FS\_SP1}}(x,y) - S_{\text{FS\_SP2}}(x,y)) \times WF_{\text{SA}}(x,y)) \quad (7)$$

$$E_{\text{ASP}} = C_1 \times \sum_{x,y \in A} (S_{\text{FS\_SP1}}(x,y) \times WF_{\text{SA}}(x,y)) - \eta \quad (8)$$

$$G_{\text{ASP}} = (E_{\text{ASP}} + \eta) / C_1. \quad (9)$$

Because  $WF$  is fixed for a specific algorithm and only  $S_{\text{FS\_SP1}}(x,y)$  may vary,  $G_{\text{ASP}}$  in (9) is a function of the MV characteristics. The MV characteristics are either the MV variances or MV standard deviations calculated based on the MVs w.r.t. a specific starting point (SP1). And the MVs are acquired by using FS on the selected sequence.

Fig. 9 shows the MV candidates that are often considered in starting point selection. They are the MVs of the neighboring spatial/temporal neighboring blocks. And the most commonly used mathematical function includes median(.) and mean(.). Combining them together, there are many possible MV predictors. In our investigation, we calculate the  $G_{\text{ASP}}$  of 36 well-known and best performed MV predictors applied to the test sequences using the weighting function of GRPS and GPHS.

We find that  $MV^{\text{pred21}}$  (mean value of  $MV^U$ ,  $MV^L$ , and  $MV^P$ ),  $MV^{\text{pred23}}$  (mean value of  $MV^U$ ,  $MV^L$  and two  $MV^P$ ) and  $MV^{\text{pred28}}$  (mean value of  $MV^{PU}$ ,  $MV^{PD}$ ,  $MV^{PL}$ ,  $MV^{PR}$ ,  $MV^{PUL}$ ,  $MV^{PUR}$ ,  $MV^{PDL}$ ,  $MV^{PDR}$ , and  $MV^P$ ) have the smallest average  $G_{\text{ASP}}$  among all the MV predictors. Together with the well-known PMV ( $MV^{\text{pred16}}$ ) and ZMV ( $MV^{\text{pred15}}$ ), these five MV predictors form the candidate set for the starting points. Note that ZMV is the abbreviation for zero motion vector (0, 0).

$$N_{\text{TSP}} = N_{\text{SPS}} + N_{\text{ASP}} - 1. \quad (10)$$

We use the initial candidate set in the following way. A proposed BME algorithm examines all MV candidates in the candidate set and then uses the best candidate as the starting point for the subsequent search procedure. The total search point number ( $N_{\text{TSP}}$ ) is shown by (10). It equals the size of starting point set ( $N_{\text{SPS}}$ ) plus the number of average search points ( $N_{\text{ASP}}$ ) produced by a specific search algorithm minus one, where “minus one” represents the initial point counted in  $N_{\text{ASP}}$ .

A well-designed starting point set should decrease  $N_{\text{ASP}}$  more than  $N_{\text{SPS}}$ , the increased size of the starting point set. We develop a systematic approach to find the optimal SPS. It is an add-on approach. At the beginning, there is only one

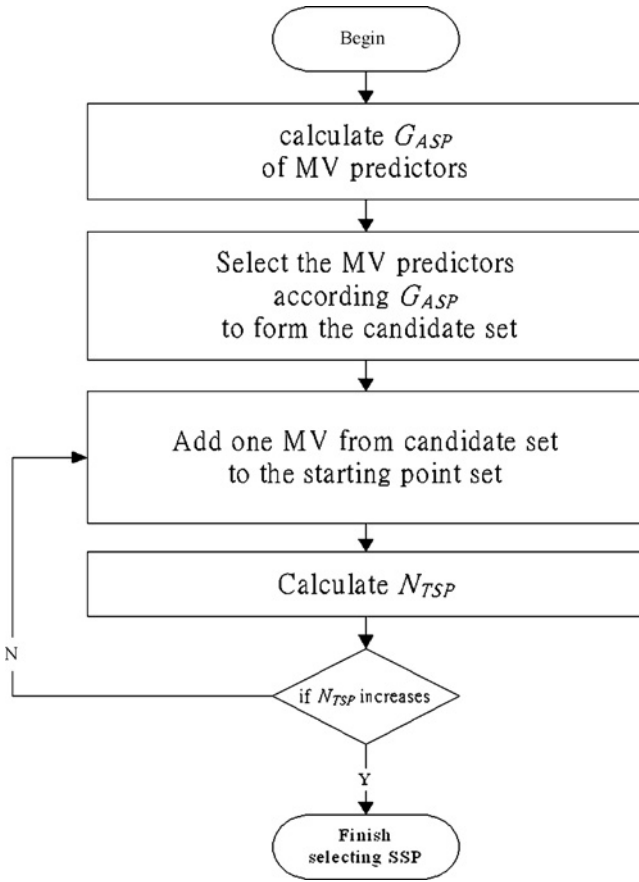


Fig. 10. Flowchart of constructing SSP.

			$SAD^{pUL}$	$SAD^{pU}$	$SAD^{pUR}$	$SAD^{UL}$	$SAD^U$	$SAD^{UR}$
	$SAD^{pp}$		$SAD^{pL}$	$SAD^p$	$SAD^{pR}$	$SAD^L$	$SAD^C$	
			$SAD^{pDL}$	$SAD^{pD}$	$SAD^{pDR}$			
Previous Previous Frame		Previous Frame			Current Frame			

Fig. 11. SAD candidates in the current frame, the previous frame, and the frame before the previous frame.

MV in the SSP. We calculate its  $N_{TSP}$  using a certain search algorithm. After a number of simulations, we retain a few best performers. We then add a second MV to each of these sets and evaluate their  $N_{TSP}$  again. We continue adding new points to each set until the  $N_{TSP}$  does not decrease with additional MV in that set. This procedure is described by the flowchart in Fig. 10. In theory, this procedure does not guarantee that the final set is globally optimal because our set is progressively constructed. However, our experiments indicate that the results are quite good.

When we apply this procedure to construct SSP for DL AGPS, the SSP for DL AGPS is  $\{PMV, MV^{pred23}\}$ . The order in the set is the order in search. The DL AGPS with SSP outperforms DL AGPS by 5% in ASP with 0.12 dB PSNR gain. Also, we observe that a fast-moving pattern

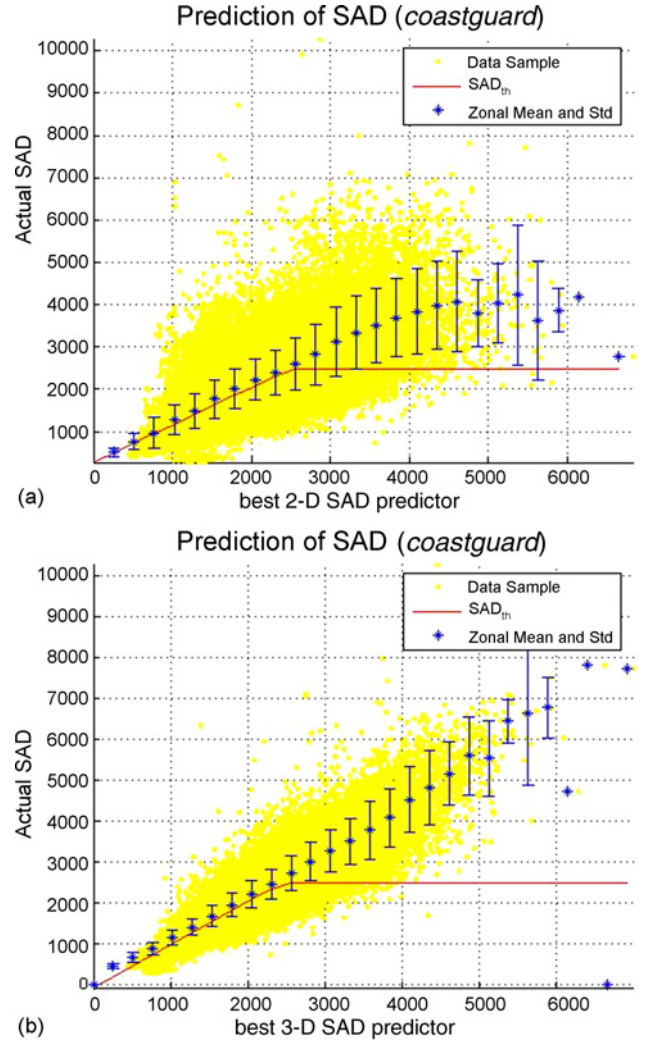


Fig. 12. Best 2-D and 3-D SAD predictor versus SADC.

search needs only a small SSP because the search algorithm can cover a large search area quickly without the help of additional starting points.

### V. EARLY TERMINATION MECHANISM

The *early termination* mechanism terminates the search process when the block-matching error produced by a MV (in the search area) is smaller than a pre-chosen threshold. And this MV is accepted as the best MV. Clearly, there is a trade-off between the MV quality (matching error) and the computational speed. Thus, the challenge is to find the termination threshold that maximizes the speed gain and minimizes the quality degradation. In this section, we set up a systematic method to find the nearly optimal early termination threshold (ETT).

The most commonly used block matching error is the sum of absolute difference (SAD). Due to the correlation among the spatial/temporal nearby blocks, [18] proposed a general form (11) of ETT. It suggests that the threshold is a function

TABLE II  
THE PERFORMANCE OF FS, DS, AIPS-MP, ARPS-ZMP, AND OUR PROPOSED ALGORITHM

Normal Sequence	FS		DS		AIPS-MP		ARPS-ZMP		Ours	
	ASP	PSNR	ASP	PSNR	ASP	PSNR	ASP	PSNR	ASP	PSNR
CT256	1024	39.56	13.81	39.51	1.63	39.64	3.67	39.56	1.36	39.63
CT40	1024	32.04	15.03	31.92	1.70	32.87	5.58	32.87	1.63	32.87
HL40	1024	33.55	15.38	34.25	2.14	34.95	5.15	34.84	1.56	35.03
MD96	1024	39.80	14.85	39.99	2.05	40.20	3.63	40.26	1.48	40.21
CG112	1024	29.08	15.09	29.14	4.06	28.89	9.93	29.07	2.23	28.99
FM512	1024	34.06	16.17	34.06	4.80	33.79	9.98	33.84	2.54	33.93
FM1024	1024	36.56	15.76	36.59	4.77	36.45	9.66	36.37	2.48	36.49
FB1024	1024	35.28	22.36	34.93	9.01	34.58	20.03	34.56	5.09	34.85
FG768	1024	26.33	15.30	26.18	3.83	26.17	7.18	26.18	3.84	26.18
ST1024	1024	29.48	16.96	29.44	5.64	29.04	11.20	28.97	2.97	29.40
Average	1024	33.57	16.07	33.60	3.96	33.66	8.60	33.65	2.52	33.76

of the SAD and the MV of the neighboring blocks

$$T = \min \left\{ \max \left\{ f \left( \begin{matrix} SAD_{1,\dots}, SAD_{i,\dots}, SAD_{n,\dots} \\ MV_{1,\dots}, MV_{i,\dots}, MV_{n,\dots} \end{matrix} \right), T_{\min} \right\}, T_{\max} \right\} \quad (11)$$

where  $SAD_i$  and  $MV_i$ , respectively, are the SAD and MV of a neighboring block labeled by  $i$ , and  $T_{\min}$  and  $T_{\max}$  stand for the lower and the upper bounds of the threshold, respectively. In practice, most researches use only the  $SAD$  predictor

$$\rho = \frac{E[(SAD^{\text{pred}} - E[SAD^{\text{pred}}]) \times (SAD^c - E[SAD^c])]}{\sqrt{E[(SAD^{\text{pred}})^2] - E^2[SAD^{\text{pred}}]} \times \sqrt{E[(SAD^c)^2] - E^2[SAD^c]}}. \quad (12)$$

To find the best threshold predictor, we use the correlation coefficient (12) between the SAD predictor ( $SAD^{\text{pred}}$ ) and the best SAD acquired using FS ( $SAD^c$ , as shown in Fig. 11) as the measure for the effectiveness of this threshold, wherein  $E[\cdot]$  represents the expected value operation. First, we perform FS on the test sequences in Table I to obtain the SAD values of all blocks. For each of the SAD predictors, we calculate its correlation with the actual SAD ( $SAD^c$ ) of the corresponding block. The one with highest correlation coefficient (closer to 1) is the best SAD predictor. By using the regression method, we find an approximation function (predictor) that best describes the relation between the predicted SAD and  $SAD^c$ . Also, we set an upper bound for the threshold estimate to prevent the quality loss in the high ETT cases. And at last, we fine-tune the predictor coefficients (slope and offset) to achieve the desired speed and quality trade-off. This fine-tuned function thus serves as the early termination threshold.

An ETT predictor often consists of two elements: (1) a selected SAD set of nearby blocks, and (2) a mathematical function operating on the selected SAD set. The most commonly used mathematical functions are mean( $\cdot$ ), median( $\cdot$ ), min( $\cdot$ ), and max( $\cdot$ ). The most commonly used 14 neighboring SADs are shown in Fig. 11. Combining them together, there are numerous possibilities. Moreover, we can insert different weights in front of each block SAD, which leads to enormous possible forms of the SAD predictors. In our letter, we select 55 representative SAD predictors and calculate the correlation coefficients between the selected SAD predictors and  $SAD^c$ . Among the 55 selected SAD predictors under consideration,  $SAD^{\text{pred}15}$  (mean SAD of the upper and left blocks) is the

best predictor in 2-D cases and  $SAD^{\text{pred}35}$  (median SAD of the upper, left, and two previous blocks) is the best predictor in all cases (2-D and 3-D cases). Herein, the 2-D cases only use the SADs of the blocks in the same frame, and the 3-D cases can also use the SADs in the previous frames.

To produce a better SAD predictor on  $SAD^c$ , we have tried the multi-dimensional regression method. But we find that the linear regression is sufficient to have a pretty accurate approximation. Consequently, (13) is the predictor of choice

$$SAD_{\text{th}}^{\text{Linear-predicted}} = K_1 \times SAD^{\text{pred}} + K_2 \quad (13)$$

where  $K_1$  and  $K_2$  are two constants divided by linear regression.

To check the effectiveness of these predictors, we calculate the mean and the standard deviation of both the best 2-D and 3-D SAD prediction errors. In Fig. 12, each dot represents the SAD pair ( $SAD^{\text{pred}}$ ,  $SAD^c$ ) of a block. The star mark at the center of a vertical bar represents the mean of  $SAD^c$ , and the bar length represents the standard deviation of prediction errors. Obviously, Fig. 12 shows that the standard deviation becomes larger when  $SAD^{\text{pred}}$  increases. This implies that for large predicted SAD values, their prediction accuracy is lower. Hence, to ensure a high MV quality, we propose an upper bound in (14) using the average SAD of all coded block in the same frame

$$SAD_{\text{th}}^{\text{Upper-bounded}} = \frac{\sum_{i=1}^{N_c-1} SAD_i}{N_c - 1} + K_3 \quad (14)$$

where  $SAD_i$  is the SAD of the  $i$ th block in the current frame,  $K_3$  is the allowed maximum early termination error offset, and  $N_c$  denotes the current block index in a frame. Finally, the early termination threshold (ETT) is defined below by (15).

$$T = SAD_{\text{th}} = \min \left( SAD_{\text{th}}^{\text{Linear-predicted}}, SAD_{\text{th}}^{\text{Upper-bounded}} \right). \quad (15)$$

The parameter values are empirically decided:  $K_1$  is set to 1,  $K_2$  is set to 384, and  $K_3$  is set to 512. Under this setting, we achieve a good balance between speed and quality.

## VI. A PBME ALGORITHM WITH ALL FEATURES

We have discussed in the previous three sections three techniques that reduce the computations of a PBME algorithm. They are (1) adaptive genetic pattern search, (2) starting point set, and (3) early termination mechanism. We now examine the performance of the PBME scheme with all the best selected tools.

The performance of FS, DS [9], ARPS-ZMP [3], AIPS-MP [4], and the DL AGPS with SPS and the 3-D ETT (our proposed best algorithm with all features) are shown in Table II. Experimental results show that the proposed algorithm outperforms ARPS-ZMP by 242% at average search points, AIPS-MP by 57%, DS by 538%, and FS by 405 times and the average PSNR quality is slightly better (0.10–0.19 dB) than all the other algorithms including FS. This may be due to the fact that our scheme often prefers a smaller value MV, which requires fewer bits in coding. And a few additional bits are available for texture (DCT coefficients) coding, which results in better overall PSNR.

## VII. CONCLUSION

In this letter, three important techniques have been investigated for reducing complexity of pattern-based block motion estimation (PBME). They are adaptive pattern search, starting point selection, and early termination. The prior arts in designing these schemes often based on heuristic reasoning and/or speculation on the collected data. The contribution of this letter is to re-examine these techniques using a systematic approach. The adopted approach is built based on analytical model and statistical tools. Optimal or nearly optimal solutions are thus proposed. Based on our previous motion estimation model and search pattern analysis [2], [10], we impose the genetic search structure on the conventional ERPS and PHS schemes to reduce computation. Furthermore, a pattern switching strategy based on the on-line MV statistics is proposed. A well-chosen starting point set indeed reduces the average number of search points. A step-by-step procedure is proposed to find the best starting point set. The so-called early termination can further improve the search speed. We suggest a metric (correlation coefficient) to identify the best predictor for determining the termination threshold. At last, a PBME algorithm combining all the above features is examined on the MPEG-4 platform. Simulations show that the search speed of the proposed algorithm is much faster than any previous search algorithm and its coding quality is kept at about the same PSNR level. All in all, the experimental results of our proposed algorithm show that our design approach for PBME is effective.

## REFERENCES

- [1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthr, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [2] J.-J. Tsai and H.-M. Hang, "Modeling of pattern-based block motion estimation and its application," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 1, pp. 108–113, Jan. 2009.
- [3] Y. Nie and K.-K. Ma, "Adaptive irregular pattern search with matching prejudgment for fast block-matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 789–794, Jun. 2005.
- [4] Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1442–1449, Dec. 2002.
- [5] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, May 2002.
- [6] C. Zhu, X. Lin, L. Chau, and L.-M. Po, "Enhanced hexagonal search for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 10, pp. 1210–1214, Oct. 2004.
- [7] L.-M. Po, C.-W. Ting, K.-M. Wong, and K.-H. Ng, "Novel point-oriented inner searches for fast block motion estimation," *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 9–15, Jan. 2007.
- [8] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 6, pp. 313–317, Jun. 1996.
- [9] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," in *Proc. Int. Conf. Inf. Commun. Signal Process. (ICICS '97)*, vol. 1, Sep. 9–12, 1997, pp. 292–296.
- [10] J.-J. Tsai and H.-M. Hang, "A genetic rhombus pattern search for block motion estimation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS '07)*, New Orleans, LA, May 2007, pp. 3655–3658.
- [11] S.-F. Lin, M.-T. Lu, H. Chen, and C.-H. Pan, "Fast multi-frame motion estimation for H.264 and its applications to complexity-aware streaming," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS '05)*, Kobe, Japan, vol. 2, May 2005, pp. 1505–1508.
- [12] S.-Y. Huang, C.-Y. Cho, and J.-S. Wang, "Adaptive fast block-matching algorithm by switching search patterns for sequences with wide-range motion content," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1373–1384, Nov. 2005.
- [13] I. Ahmad, W. Zheng, J. Luo, and M. Liou, "A fast adaptive motion estimation algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 420–438, Mar. 2006.
- [14] M.-C. Chang and J.-S. Chien, "An adaptive search algorithm based on block classification for fast block motion estimation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS '06)*, May 2006, pp. 3982–3985.
- [15] Y. Liu and S. Orintara, "Complexity comparison of fast block-matching motion estimation algorithms," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP '04)*, vol. 3, May 2004, pp. 341–344.
- [16] I. Gonzalez-Diaz and F. Diaz-de-Maria, "Adaptive multipattern fast block-matching algorithm based on motion classification techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 10, pp. 1369–1382, Oct. 2008.
- [17] J.-J. Tsai and H.-M. Hang, "On adaptive pattern selection for block motion estimation algorithms," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP '07)*, Honolulu, HI, vol. 1, Apr. 2007, pp. I-1173–I-1176.
- [18] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive motion vector field adaptive search technique (PMVFAST): Enhancing block based motion estimation," in *Proc. Vis. Commun. Image Process. (VCIP '01)*, Jan. 2001, pp. 883–892.
- [19] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Proc. Visual Commun. Image Process. (VCIP '02)*, San Jose, CA, Jan. 2002, pp. 1069–1079.
- [20] J.-J. Tsai and H.-C. Chen, "Predictive block-matching discrepancy based rhombus pattern search for block motion estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP '05)*, Genova, Italy, vol. 1, Sep. 2005, pp. 1073–1076.
- [21] C.-H. Lin and J.-L. Wu, "A lightweight genetic block-matching algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 386–392, Aug. 1998.
- [22] M. F. So and A. Wu, "Four-step genetic search for block motion estimation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP '98)*, vol. 3, May 1998, pp. 1393–1396.