

Modeling of Pattern-Based Block Motion Estimation and Its Application

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

Abstract—Pattern-based block motion estimation (PBME) is one of the most widely adopted compression tools in the contemporary video coding systems. However, despite that many researches have studied PBME, few have yet attempted to construct an analytical model that can explain the underneath principle and mechanism of various PBME algorithms. In this paper, we propose a statistical PBME model that consists of two components: 1) a statistical probability distribution for motion vectors and 2) the minimal number of search points (so-called weighting function) achieved by a search algorithm. We first verify the accuracy of the proposed model by checking the experimental data. Then, an application example using this model is shown. Starting from an ideal weighting function, we devise a novel genetic rhombus pattern search (GRPS) to match the design target. Simulations show that, comparing to the other popular search algorithms, GRPS reduces the average search points for more than 20% and, in the meanwhile, it maintains a similar level of coded image quality.

Index Terms—Genetic search, modeling, motion estimation, pattern-based block motion estimation (BME), video coding.

I. INTRODUCTION

BLOCK MOTION estimation (BME) is a critical component in an efficient inter-frame coding. However, it is a highly computation-intensive process. Many researches have proposed fast algorithms to reduce its computational requirement. However, few researchers, to our knowledge, have tried to construct an accurate model for the BME process. To be specific, it is a model that unveils the relationship among the video sequences, the search methods, the computational complexity and the output video quality. Our aim is to construct an explicit mathematical model for BME.

According to [1], the fast BME algorithms can be classified mainly into two categories: 1) reducing of number of checking (search) points and 2) lowering computational complexity in calculating the block-matching criterion for each checking (search) point. This study focuses on the algorithms in the first category. The first fast BME category roughly consists of three sets of tools for reducing the search points: 1) an operative threshold for early decision mechanisms [2]–[4], 2) the selection of good initial search points [3], [5], and 3) an effective set of search patterns [4], [6]–[8]. Combing all these tools, the latest BME

algorithms achieve a dramatic speed-up in finding the near-optimal candidate motion vectors while maintaining a desired level of quality. The first and second sets of tools rely heavily on the data dependency of neighboring image data. Consequently, the search pattern plays a key role in deciding the performance of a search algorithm especially when the data correlation is low. In this paper, we like to explore the following problems. Why does one search pattern outperform the others? What is the underlying mechanism behind it? Is there a search pattern that handles nearly all sequences well? Moreover, can we construct a mathematical model that describes the underlying mechanism? An attempt is made in this paper to answer these questions.

In this paper, we will construct a simple and yet effective statistical model for the pattern-based BME (PBME). Also, based on this model, a novel genetic PBME algorithm is devised. The rest of this paper is organized as follows. Section II presents the probability distribution functions of the motion vectors acquired by full search (FS). In Section III, we analyze the search points of several representative PBME algorithms and formulate the weighting functions (*WF*). Based on the proposed probability distribution function for motion vectors and the *WFs* of different PBME algorithms, Section IV constructs a statistical model for PBME. To demonstrate the usefulness of this model, a new genetic rhombus pattern search is presented in Section V, which shows good performance for both low motion and relative high motion sequences. Lastly, we conclude this paper by Section VI.

II. PROBABILITY DISTRIBUTION FOR MOTION VECTORS

In order to design a good search pattern set, many papers discussed the nature of motion vectors (MVs). The authors of [9] empirically gather the statistics of the motion vectors around the initial search point. And [10] assumes that the motion vector distribution can be approximated by either Gaussian or Laplacian probability distributions. So far, we have not found an attempt of finding a probability distribution function (PDF) that has a very precise match to the motion vectors.

We select a few representative training sequences to generate motion vectors at various bit rates under the settings given in Table I. These sequences are coded by a MPEG-4 SP@L3 encoder using FS. All the sequences are in CIF (352×288) format. Only the first frame is coded as I frame, and all the remaining frames are coded as P frames. The motion vector search range is set to 16, the initial quantizer step size is set to 15, and the block size is set to 16×16 . When the quantization step varies to achieve the desired bit rate, the peak signal noise ratio (PSNR) quality of the coded video sequence ranges from 26dB (poor but acceptable) to 40dB (visually the same as original).

A. Motion Vector Distributions

In our experiments, we test two kinds of initial motion vectors (served as the origins of the PBME search), namely, the zero

Manuscript received August 03, 2006; revised August 26, 2007. First published December 09, 2008; current version published January 21, 2009. This paper was recommended by Associate Editor T. Wiegand.

J.-J. Tsai is with Department of Electronics Engineering, National Chiao-Tung University, Hsinchu 30010, Taiwan (e-mail: jtsai@aa.nctu.edu.tw).

H.-M. Hang is with Department of Electronics Engineering, National Chiao-Tung University, Hsinchu 30010, Taiwan, and Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 10061, Taiwan (e-mail: hmhang@mail.nctu.edu.tw).

Corresponding Author: H.-M. Hang, Department of Electronics Engineering, National Chiao-Tung University, Hsinchu 30010, Taiwan (e-mail: hmhang@mail.nctu.edu.tw).

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

Digital Object Identifier 10.1109/TCSVT.2008.2009248

TABLE I
 TRAINING SEQUENCES AND THEIR CODING PARAMETERS

Label	Sequence	Bit rate (K bps)	Frame rate (fps)	Number of frames	PSNR
CT256	container	256	7.5	300	39
CT40	container	40	7.5	300	32
HL40	hall	40	7.5	300	33
MD96	mother and daughter	96	10	300	40
CG112	coastguard	112	30	300	29
FM512	foreman	512	30	300	34
FM1024	foreman	1024	30	300	36
FB1024	football	1024	30	90	35
FG768	flower garden	768	30	250	26
ST1024	Steven	1024	30	300	29

motion vector (ZMV) and the predicted motion vector (PMV). Herein, ZMV is defined by (1), and PMV is defined by (2), which is specified by the MPEG-4 standard.

$$\text{ZMV} = (0, 0) \quad (1)$$

$$\text{PMV} = \text{Median}(\text{MV}^U, \text{MV}^L, \text{MV}^{UR}) \quad (2)$$

where MV^U is the adjacent upper block of the current block, MV^L is the adjacent left block, and MV^{UR} is the neighboring up-right block.

Based on the motion vector data obtained by applying FS to video sequences, we find that the motion vector distributions with respect to (w.r.t.) PMV generally have a more symmetric shape as compared to the motion vector distributions w.r.t. ZMV. In addition, the PMV-based motion vectors have a much smaller standard deviation. They cluster better. Therefore, the motion vector distributions w.r.t. PMV are used in the rest of this paper. The statistics of the motion vectors w.r.t. PMV of all the selected training sequences show that both the horizontal mean values and vertical mean values are close to zero. Thus, these motion vector distributions are zero-biased w.r.t. PMV. Furthermore, the correlations between the horizontal components and the vertical components of motion vectors are nearly zero for all our training sequences.

B. Normalized Independent 2-D Distribution

Based on the above observations, three popular zero-mean normalized independent 2-D distributions are considered as candidates for modeling the MV distribution: 1) Gaussian distribution function (3), 2) Laplacian distribution function (4), and 3) Cauchy distribution function (5). Note that $(x, y) \in A$, $(x', y') \in A$ and A is the geographical area of $[-32 \sim +31, -32 \sim +31]$ in our experiments. The parameters $(\lambda_x, \lambda_y, b_x, b_y, \eta_x$ and $\eta_y)$ are functions of the MV variances

$$G(x, y) = \frac{e^{-x^2/2\lambda_x} \cdot e^{-y^2/2\lambda_y}}{\sum_{(x', y') \in A} e^{-x'^2/2\lambda_x} \cdot e^{-y'^2/2\lambda_y}} \quad (3)$$

$$L(x, y) = \frac{e^{-|x|/b_x} e^{-|y|/b_y}}{\sum_{(x', y') \in A} e^{-|x'|/b_x} e^{-|y'|/b_y}} \quad (4)$$

$$C(x, y) = \frac{\frac{1}{x^2 + \eta_x^2} \frac{1}{y^2 + \eta_y^2}}{\sum_{(x', y') \in A} \frac{1}{x'^2 + \eta_x^2} \frac{1}{y'^2 + \eta_y^2}} \quad (5)$$

Remark: Strictly speaking, the zero correlation between the horizontal components and vertical components of motion vectors does not imply that they are statistically independent. However, we justify the correctness of these probabilistic models using the goodness-of-fit test [11] as follows.

To find out which of the three PDFs best approximates the PDF of motion vectors acquired by FS, a well-known goodness-of-fit test, 2-D KS test [12], is adopted. The statistic D defined in [12] is used as the measure of similarity between the hypothesized PDF (the modeled data) and the observed PDF (measured data). To be more specific, the statistic D is the maximum absolute difference between two cumulative probability distributions function, as defined in (6), wherein $\text{CDF}_{\text{modeled}}$ and $\text{CDF}_{\text{measured}}$ are the cumulative PDFs of the model and the measured data, respectively, and A is the geographical area of $[-32 \sim +31, -32 \sim +31]$ in our experiments

$$D = \max \{ |\text{CDF}_{\text{modeled}}(v) - \text{CDF}_{\text{measured}}(v)| \}, v \in A. \quad (6)$$

A smaller statistic D implies that the hypothesized PDF matches better the observed PDF. The motion vector probability distributions acquired by FS, PDF_{FS} , are tested against the aforesaid hypothesized zero-mean, normalized, independent 2-D distributions with the same variances. These hypothesized distributions with the same variance of MV acquired by FS are called $G_{\text{FS}}(x, y)$, $L_{\text{FS}}(x, y)$ and $C_{\text{FS}}(x, y)$, respectively. In our experiments, $C_{\text{FS}}(x, y)$ generally has the smallest statistic D values. However, according to [12], the values of statistic D in our experiments are too large to claim that any of these three 2-D distributions has a good match to the target $\text{PDF}_{\text{FS}}(x, y)$.

C. A Fitted Probability Distribution

To construct a more accurate PDF model, we extend $C(x, y)$ and propose a new form of PDF denoted by $T(x, y)$, which is defined by (7). For each of the selected training sequences, τ_x and τ_y are optimized such that the maximum discrepancy between $\text{PDF}_{\text{FS}}(x, y)$ and $T(x, y)$ is minimized, and ξ_x and ξ_y are adjusted such that the variances of $T(x, y)$ are the same as those of the training sequences. The $T(x, y)$ with the fitted parameters that match $\text{PDF}_{\text{FS}}(x, y)$ becomes $T_{\text{FS}}(x, y)$. Experiments show that τ_x and τ_y range from 1.13 to 2.2 for the training sequences. This indicates the variations among the training sequences are considerably large

$$T(x, y) = \frac{\frac{1}{|x|^{\tau_x + \xi_x}} \frac{1}{|y|^{\tau_y + \xi_y}}}{\sum_{(x', y') \in A} \frac{1}{|x'|^{\tau_x + \xi_x}} \frac{1}{|y'|^{\tau_y + \xi_y}}} \quad (7)$$

Despite the large individual differences among the training sequences, we find that τ_x and τ_y are generally around 1.67. We

TABLE II
STATISTIC D OF 2-D KS TEST ON SOME REPRESENTATIVE SEQUENCES

Sequences	$G_{FS}(x,y)$	$L_{FS}(x,y)$	$C_{FS}(x,y)$	$S_{FS}(x,y)$
CT256	0.48	0.38	0.08	0.02
CT40	0.42	0.35	0.14	0.06
MD96	0.38	0.32	0.12	0.08
FB1024	0.28	0.23	0.19	0.14
ST1024	0.39	0.33	0.17	0.11
Average	0.39	0.32	0.14	0.08

thus choose $\tau_x = \tau_y = 5/3$ to simplify $T(x, y)$. The resultant distribution is called $S(x, y)$ as defined by (8)

$$S(x, y) = \frac{\frac{1}{|x|^{5/3} + \zeta_x} \frac{1}{|y|^{5/3} + \zeta_y}}{\sum_{(x', y') \in A} \frac{1}{|x'|^{5/3} + \zeta_x} \frac{1}{|y'|^{5/3} + \zeta_y}}. \quad (8)$$

In Table II, the 2-D KS tests show that $S_{FS}(x, y)$ has a smaller statistic D in comparison with $G_{FS}(x, y)$, $L_{FS}(x, y)$, and $C_{FS}(x, y)$. Note that the parameters (ζ_x, ζ_y) of $S_{FS}(x, y)$ are obtained by numerical methods so that the variances of $S_{FS}(x, y)$ match the data statistics of motion vectors acquired by FS. In summary, we propose a new probability distribution $S(x, y)$ that models the PDF of the motion vectors. It constitutes the first element of our complete PBME model.

III. SEARCH POINTS IN PATTERN-BASED SEARCH ALGORITHMS

Search patterns are generally devised based on the assumption that the matching cost surface is uni-modal; in other words, the matching cost associated with a search point is smaller when it is closer to the global minimum. Under this assumption, the number of search points is defined to be the minimal number of search points in all possible paths leading to the best-matched point from the starting (initial) point. Because the (shortest) search path is determined by the search pattern in general, the search point number depends on the search pattern. Therefore, it (number of search points) is a discrete function of the location and is called weighting function (WF). By examining the search process of a PBME, we can construct its corresponding WF. Note that the global uni-modal cost surface assumption is too strong and it is not always valid for typical video sequences [6]. Often it is valid within a small neighborhood of the global minimum point. Consequently, the WF does not represent the actual number of search points. To be exact, it represents the lower bound of the number of search points. But the statistics also show that the number of actual search points is highly correlated with our defined WF.

Four representative pattern-based search methods, four step search (FSS) [7], diamond search (DS) [8], enhanced hexagonal search (EHS) [6], and easy rhombus pattern search (ERPS), are used to illustrate the construction of weighting functions. Herein, ERPS is ARPS [4] without searching various MV predictors and it uses PMV as the starting point. These pattern-based search algorithms are chosen because of their well-recognized performance.

Fig. 1 shows the contour plots of the WFs of FSS, DS, EHS, and ERPS, respectively. The value on a contour represents the

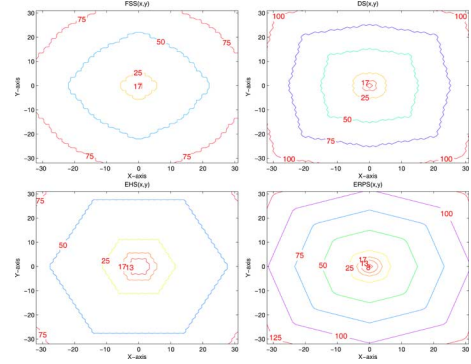


Fig. 1. Contour plots of the WFs of FSS, DS, EHS, and ERPS, respectively.

least number of search points for a search algorithm to move from the origin to a point (location) on the contour. Because EHS moves faster than any other algorithms, EHS surpasses the other algorithms at distant locations. Its weighting function $WF_{EHS}(x, y)$ has smaller values at the outer contour. On the other extreme, because $WF_{ERPS}(x, y)$ has the smallest values around the starting point, ERPS has advantages in slow motion situations. Therefore, by looking into the WF of a search algorithm, we understand why one search algorithm works better for a particular situation (fast motion or slow motion). WF is the second element of our complete PBME model.

IV. STATISTICAL MODEL FOR PATTERN-BASED BLOCK MOTION ESTIMATION

Based on the problem formulation in Sections II and III, the average search points (ASP) of a PBME algorithm for a sequence can be described by (9). It depends on both search algorithm (SA) and the video sequence. Mathematically, it is the sum of the products of WF and the MV PDF at all locations within the search area, where $SPF_{SA}(x, y)$ denotes the number of search points acquired by a specific algorithm, $PDF_{SA}(x, y)$ denotes the motion vector distribution acquired by a specific algorithm on a specific sequence, and A is the search area. It is clear that we can get $SPF_{SA}(x, y)$ and $PDF_{SA}(x, y)$ only after we apply a specific search algorithm to a specific sequence.

In this paper, we propose to use (10) for modeling ASP. In (10), ASP is a linear function of the sum of the products of $S_{FS}(x, y)$ and $WF_{SA}(x, y)$. Thus, it consists of two components: MV distribution (sequence dependent) and WF (SA dependent). By tuning the values of C_1 and C_2 , we can compensate for the modeling errors. Given a SA, we pre-analyze it and obtain $WF_{SA}(x, y)$. Given a specific sequence, we can pre-calculate MVs using FS and obtain $S_{FS}(x, y)$. Then, one can use (10) to estimate the ASP values of an SA when it is applied to a specific sequence

$$ASP = \sum_{(x,y) \in A} PDF_{SA}(x, y) \times SPF_{SA}(x, y) \quad (9)$$

$$ASP = C_1 \times \sum_{x,y \in A} S_{FS}(x, y) \times WF_{SA}(x, y) + C_2. \quad (10)$$

We need to justify that the above model is valid for real data. There are two methods to decide C_1 and C_2 . In the first method, we apply a fixed SA to a set of training sequences to compute

TABLE III
REGRESSION PARAMETERS (C_1 AND C_2) AND THE CORRELATION COEFFICIENTS BETWEEN THE MODEL-PREDICTED ASP AND THE TRUE ASP. (1ST METHOD)

BME	C_1	C_2	ASP correlation
FSS	0.42	10.38	0.98
DS	0.46	7.59	0.98
EHS	0.42	5.63	0.99
ERPS	0.44	2.97	0.98

TABLE IV
REGRESSION PARAMETERS (C_1 AND C_2) AND THE CORRELATION COEFFICIENTS BETWEEN MODEL-PREDICTED ASP AND THE TRUE ASP. (2ND METHOD)

Sequence	C_1	C_2	ASP correlation
CT256	1.07	-1.42	1.00
CT40	1.17	-4.70	0.98
HL40	1.19	-4.35	0.99
MD96	1.17	-4.52	0.97
CG112	1.05	-1.05	1.00
FM512	1.15	-3.60	0.99
FM1024	1.10	-2.36	1.00
FB1024	0.62	1.66	0.73
FG768	1.15	-3.76	0.98
ST1024	1.08	-5.82	0.91

C_1 and C_2 by the regression method. Our objective is that the model with trained C_1 and C_2 can predict the ASP of a new sequence accurately. In the second method, we apply a few search algorithms (the training algorithms) to a specific sequence, and then calculate C_1 and C_2 based on the acquired data. In this case the goal is that the model with trained C_1 and C_2 can predict the ASP values of a new algorithm.

In the first method, C_1 and C_2 are obtained from a set of training sequences with one specific search algorithm. Table III displays the C_1 and C_2 values for each search algorithm. The last column is the correlation coefficient between the actual ASP and the predicted ASP. One may notice that the correlation coefficients are all very close to 1, which indicates that the predicted ASPs are nearly the same as the actual ASPs.

In the second method, C_1 and C_2 are obtained by applying a set of search algorithms (training algorithms) to a specific sequence. We then predict the ASP value of a new algorithm by using the proposed model. Table IV displays the C_1 and C_2 values for the 10 sequences and the correlation coefficients between the predicted ASP and the actual ASP. The correlation coefficients are very close to 1 except that for the FB1024 sequence. This may be due to the high motion contents of FB1024. In spite of the small number of training algorithms, the coherence between the predicted ASP and the actual ASP is very high for all 10 sequences.

The first method and the second method are designed for different scenarios. The first method is used to predict the ASP of a new sequence (for a given specific search algorithm), while the second method is used to predict the ASP of a new search algorithm (for a given specific sequence). Due to different sizes of training samples and different purposes, the accuracy comparison between these two methods may not be meaningful.

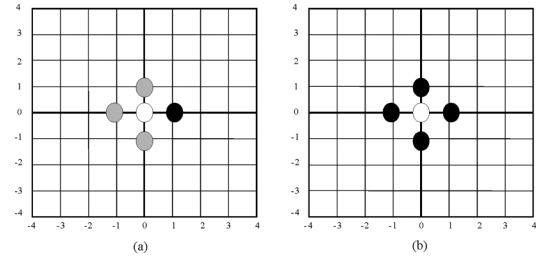


Fig. 2. Search patterns for GRPS.

V. AN APPLICATION: PATTERN-BASED SEARCH ALGORITHM DESIGN

How can we devise a new pattern-based search algorithm with the help of the previous analysis? We do this in three steps. We first construct a target WF based on the analysis in the past two sections. Then, we devise a search pattern that hopefully achieves the desired WF. At last, we evaluate its performance by simulation on real pictures.

The first step of designing a new search algorithm is to find a WF that has the smallest possible values at all locations, because, in our proposed model, WF is the only algorithm-dependent parameter that determines ASP.

Most effective PBME algorithms consist of two stages: 1) coarse regular search stage and 2) fine ending search stage. The purpose of the regular search stage is to fast locate the potential optimal motion vectors, and the ending stage is to determine the best-matched point in a small neighborhood. Each stage may use one or several search patterns. In the regular search stage, because the shortest path between two points in a plane is a straight line, the fastest search path for a search algorithm is the straight line from the starting point directly to the best-matched motion vector. Based on the previous experiments, we suspect that a doable search method moves at most one unit distance horizontally or vertically per step. As shown in Fig. 2(a), the minimal number of search points for reaching the motion vector (x, y) is “ $abs(x) + abs(y) + 1$.” In the ending stage, to decide precisely the location of the best candidate motion vector generally requires to search at least the neighboring 4 points and the current point (center) itself, as shown in Fig. 2(b). The resulting contour plot of $WF_{GRPS}(x, y)$ is depicted in Fig. 3.

The second step is to choose proper search patterns that fulfill the desired WF. By simplifying the genetic search algorithms in [13] and [14] and combining the rhombus search patterns, we propose a genetic rhombus pattern search (GRPS) algorithm described below.

- 1) *Initialization*: Check the starting point, PMV, and set it as the parent point.
- 2) *Mutation*: Randomly select a next generation point (the mutation point) from the untested points of a rhombus pattern centered at the parent. (That is, check one of the four solid points in the coarse search pattern in Fig. 2(a).)
- 3) *Competition*: Select the survivor between the parent and its mutation based on their matching costs.
 - a) If the mutation is better than the parent, the mutation is the survivor (the next parent). Go to step 2.

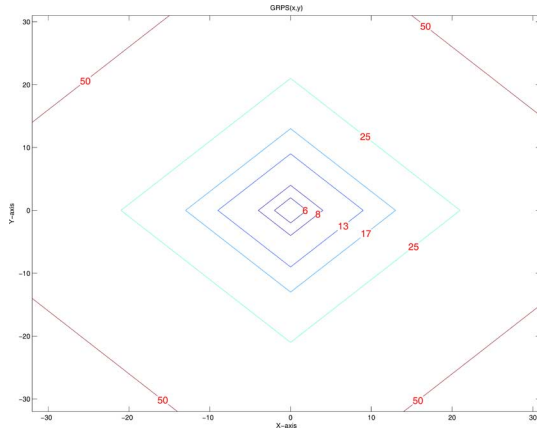


Fig. 3. WF of GRPS.

- b) If the parent is better than its mutation, the parent is the survivor (the next parent) and check if there is any remaining untested mutation point in the four points of a rhombus pattern. If there is one, go to step 2; otherwise, (that is, all points in the ending search pattern, Fig. 2(b), are checked,) go to step 4.

4) *End*: Set the current survivor as the final motion vector.

Comparing Fig. 3, the contour plot of $WF_{GRPS}(x, y)$, with the contour plots of the four popular algorithms in Fig. 1, we can find that GRPS has the same or smaller number of search points as ERPS near the starting point, and it has a smaller number of search points than EHS in locations away from the starting point. In other words, it achieves the smallest number of search points at nearly all locations, when compared to the four popular search algorithms.

Lastly, we evaluate the performances of the proposed GRPS by conducting experiments on the selected training sequences. The results are shown in Table V (average number of search points), Table VI (PSNR), and Table VII (performance comparison). In Table VII, the computing gain (CG) is defined as the ratio of ASP minus one, and the quality gain (QG) is defined as the PSNR difference. In summary, the ASP of GRPS on average is 22% faster than that of ERPS, 56% faster than EHS, 130% faster than DS, 172% faster than FSS, and 145 times faster than FS. On the other hand, the PSNR of GRPS is on average better than all other search algorithms, except for ERPS. Compared with ERPS, the quality loss of GRPS is very small, around 0.01dB. Therefore, GRPS outperforms all the other search algorithms in terms of ASP for all training sequences, and its coding quality is comparable with all the other algorithms.

The GRPS algorithm has fewer ASP because the proposed genetic pattern search calculates and compares the matching cost of the parent with that of one randomly selected point in the search pattern. On the average, the genetic algorithm saves about 50% search points when the matching error surface is nearly monotonic. In the worst cases, its behavior is the same as the non-genetic sibling. Considering the probability of being trapped into the local minimum, a genetic algorithm roughly has a similar behavior as its non-genetic sibling. The reason is that both of them terminate the search process when the matching error of the center point is smaller than those of all points in the

TABLE V
ASP (AVERAGE NUMBER OF SEARCH POINTS)

ASP	GRPS	ERPS	EHS	DS	FSS	FS
CT256	5.36	5.75	9.59	13.81	17.53	1024
CT40	5.98	7.04	10.42	15.03	18.38	1024
HL40	6.35	7.33	10.34	15.38	18.72	1024
MD96	5.98	6.83	10.32	14.85	18.37	1024
CG112	6.08	7.63	10.31	15.09	18.25	1024
FM512	7.13	8.65	10.76	16.17	19.03	1024
FM1024	6.94	8.32	10.54	15.76	18.71	1024
FB1024	11.89	16.36	14.29	22.36	22.70	1024
FG768	6.38	7.57	10.55	15.30	18.73	1024
ST1024	7.65	9.95	11.48	16.96	19.47	1024
Average	6.97	8.54	10.86	16.07	18.99	1024

TABLE VI
PSNR (PEAK SIGNAL NOISE RATIO)

PSNR	GRPS	ERPS	EHS	DS	FSS	FS
CT256	39.49	39.50	39.48	39.51	39.49	39.56
CT40	32.21	32.08	31.46	31.92	31.69	32.04
HL40	34.49	34.60	34.27	34.25	34.17	33.55
MD96	40.08	40.09	39.87	39.99	39.93	39.80
CG112	29.14	29.16	29.07	29.14	29.13	29.08
FM512	34.05	34.10	33.94	34.06	34.02	34.06
FM1024	36.52	36.61	36.46	36.59	36.48	36.56
FB1024	34.87	34.88	34.86	34.93	34.94	35.28
FG768	26.17	26.19	26.15	26.18	26.16	26.33
ST1024	29.39	29.31	29.47	29.44	29.35	29.48
Average	33.64	33.65	33.50	33.60	33.54	33.57

TABLE VII
CODING PERFORMANCE COMPARISON

Gain	GRPS over ERPS		GRPS over EHS		GRPS over DS		GRPS over FSS		GRPS over FS	
	CG	QG	CG	QG	CG	QG	CG	QG	CG	QG
CT256	0.07	-0.01	0.79	0.02	1.58	-0.01	2.27	0.00	190.04	-0.07
CT40	0.18	0.13	0.74	0.74	1.51	0.28	2.07	0.51	170.24	0.16
HL40	0.15	-0.11	0.63	0.22	1.42	0.24	1.95	0.32	160.26	0.94
MD96	0.14	-0.02	0.73	0.20	1.48	0.08	2.07	0.15	170.24	0.27
CG112	0.25	-0.02	0.70	0.07	1.48	0.00	2.00	0.01	167.42	0.06
FM512	0.21	-0.05	0.51	0.12	1.27	-0.01	1.67	0.03	142.62	-0.00
FM1024	0.20	-0.08	0.52	0.07	1.27	-0.06	1.70	0.04	146.55	-0.04
FB1024	0.38	-0.01	0.20	0.01	0.88	-0.06	0.91	-0.06	85.12	-0.41
FG768	0.19	-0.02	0.65	0.02	1.40	-0.00	1.94	0.01	159.50	-0.15
ST1024	0.30	0.07	0.50	-0.08	1.22	-0.06	1.55	0.04	132.86	-0.09
Average	0.22	-0.01	0.56	0.14	1.30	0.04	1.72	0.11	145.83	0.07

search pattern. But they may be trapped into different locations. We have examined the motion vectors produced by GRPS and by the other search algorithms. Using the MVs produced by FS as the ideal target, the average absolute differences in MVs are similar for the five fast SAs (including GRPS) discussed in this paper. Also, the average SAD differences of these five SAs are similar.

VI. CONCLUSION

A systematic approach is taken in constructing a mathematical model for the PBME algorithms. With the assistance of goodness-of-fit tests, we propose a new PDF, an extended

Cauchy distribution, for the motion vector distribution. It matches well the real motion vector PDF produced by FS. We then suggest a so-called WF that describes the minimal search points of a search algorithm. The WF of a certain PBME algorithm is estimated by analyzing the search process of that PBME. The complete PBME model includes these two elements: the statistical PDF derived from a video sequence and the WF derived from a search algorithm. With the proposed model, we can predict the performance of a new search pattern without actually applying the search algorithm to a video sequence. Thus, it helps us in constructing new search patterns (algorithms). An application example is given. Starting from an ideal WF target, we propose a GRPS algorithm, which outperforms all other popular search algorithms in speed while maintaining a similar PSNR quality.

REFERENCES

[1] C. Zhu, W. S. Qi, and W. Ser, "Predictive fine granularity successive elimination for fast optimal block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 213–221, Feb. 2005.

[2] P. I. Hosur and K.-K. Ma, "Motion vector field adaptive fast motion estimation," presented at the 2nd Int. Conf. ICICS, Singapore, Dec. 1999.

[3] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive motion vector field adaptive search technique (PMVFAST) – Enhancing block based motion estimation," *Proc. VCIP*, pp. 883–892, Jan. 2001.

[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. 1142–1449, Dec. 2002.

[5] A. M. Tourapis, O. C. Au, and M. L. Liou, "New results on zonal based motion estimation algorithms-advanced predictive diamond zonal search," in *Proc. IEEE ISCAS*, May 2001, vol. 5, pp. 183–186.

[6] C. Zhu, X. Lin, L. P. Chau, and L. M. Po, "Enhanced hexagonal search for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 10, pp. 1210–1214, Oct. 2004.

[7] 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. 3, pp. 313–317, Jun. 1996.

[8] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," in *Proc. ICICS*, Sep. 9–12, 1997, vol. 1, pp. 292–296.

[9] C. H. Cheung and L.-M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 16–22, Feb. 2005.

[10] Y. Liu and S. Orintara, "Complexity comparison of fast block-matching motion estimation algorithms," in *Proc. IEEE ICASSP*, May 2004, vol. 3, p. iii-341–4.

[11] S. D. Silvey, *Statistical Inference*. London, U.K.: Chapman Hall, 1975.

[12] J. Peacock, "Two-dimensional goodness-of-fit testing in astronomy," *Monthly Notices of the Royal Astronomical Society*, vol. 202, pp. 615–627, 1983.

[13] 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.

[14] M. F. So and A. Wu, "Four-step genetic search for block motion estimation," in *Proc. IEEE ICASSP*, May 1998, vol. 3, pp. 1393–1396.