A Fast Adaptive Motion Estimation Algorithm

Ishfaq Ahmad, *Senior Member, IEEE*, Weiguo Zheng, *Member, IEEE*, Jiancong Luo, *Member, IEEE*, and Ming Liou, *Life Fellow, IEEE*

***Abstract*—Motion estimation (ME) is a multistep process that involves not one, but a combination of techniques, such as motion starting point, motion search patterns, and adaptive control to curb the search, avoidance of search stationary regions, etc. The collective efficiency of these techniques is what makes a ME algorithm robust and efficient across the board. This paper proposes a ME algorithm that is an embodiment of several effective ideas for finding the most accurate motion vectors (MVs) with the aim to maximize the encoding speed as well as the visual quality. The proposed algorithm takes advantage of the correlation between MVs in both spatial and temporal domains, controls to curb the search, avoids of search stationary regions, and uses switchable shape search patterns to accelerate motion search. The algorithm yields very similar quality compared to the full search but with several hundred times faster speed. We have evaluated the algorithm through a comprehensive performance study that shows that the proposed algorithm achieves substantial speedup without quality loss for a wide range of video sequences, compared with the ME techniques recommended by the MPEG-4 committee.**

***Index Terms*—Adaptive search patterns, adaptive threshold, inertia tracing, spatial and temporal predictive search.**

I. Introduction

THE overwhelming complexity of motion estimation (ME) using a full search based brute-force approach has led to explosive research in ME. The research has led to myriad fast algorithms, and yet finding the "most efficient" algorithm remains an open research problem. Most ME algorithms exhibit tradeoffs between quality and speed. Since ME is highly scene dependent, and, no one technique can be fully relied to generate good visual quality for all kinds of video scenes. Instead, it is the quintessence of a variety of techniques, such as motion starting point, motion search patterns, and adaptive control to curb the search, avoidance of search stationary regions, etc., that makes an ME algorithm robust and efficient across the board.

While the performance of the full search method is considered to be "optimal," its complexity is prohibitively high for software implementation. Furthermore, since the full search method aims to find the minimum sum of absolute differences (SADs), the presence of noise in a video can lead to suboptimal motion vectors (MVs). The presence of noise can also cause the full search to produce chaotic motion field for a smooth motion video, costing more bits to encode MVs with fewer bits left for encoding DCT coefficients with a given bit budget [1], [22]. Designing fast and accurate ME algorithms remains an open research problem. The most common ME method is the block matching technique, in which a video frame is divided into macroblocks (MBs) (16×16 pixels) or blocks (8×8 pixels) and a search window is defined. Each MB of the current frame is compared with the blocks of the reference frame within a search window. The displacement with the maximum correlation or the minimum distortion between the current block and the reference blocks within the search window is selected as the MV.

A vast number of block matching algorithms (BMAs) have been proposed (see [18] for an extensive survey). Some of the well known algorithms are: block pixel decimation [16], three step search algorithm (TSS) [18] and two dimensional logarithmic search algorithm (2D-LOG) [16] as well as their variations [25], [21], new three step search algorithm (NTSS) [22], four step search algorithm [26], conjugate directional search [23], [31], [37], orthogonal direction search algorithm (OSA) [27], dynamic search window adjustment (DSWA) [20], cross search [7], diamond search [32], [39], gradient-based search [24], zone-based search [17], refined zone-based search [11], parallel hierarchical one-dimensional search algorithm (PHODS) [4], and candidate vector-based four step search algorithm (CV4SS) [12]. Hierarchical search [3], [5], [21], [29] and multiresolution algorithms [2], [3] perform ME at multiple levels successively, starting with the lowest resolution level using low-pass filtering or subsampling [28].

Almost all of the BMAs make explicit and implicit assumptions that the matching distortion increase monotonically as the checking point moves away from the global minimum or the error surface is unimodal over the global window. Indeed, this assumption is not always true. Consequently, the resultant MVs may be trapped in a local minimum. Most BMAs exhibit proper behavior provided the following prerequisites are met: 1) object displacement is constant within a block of pixels; 2) pixel illumination between successive frame is spatially and temporally uniform (this constraint can be relaxed in the BMA with luminance correction); 3) motion is restricted to translation; 4) matching distortion increases monotonically as the displaced candidate block moves away from the direction of the exact minimum distortion. Most of these conditions are usually not met for real-life video sequences, but a large number of ME algorithms still perform reasonably well. The problem is these algorithms yield different performance on different

Manuscript received February 22, 2003; revised August 28, 2004. This paper was recommended by Associate Editor S.-U. Lee.

I. Ahmad is with the Computer Science and Engineering Department, University of Texas, Arlington, TX 96019 USA (e-mail: iahmad@cse.uta.edu).

W. Zheng was with Hong Kong University of Science and Technology, Hong Kong, SAR. He is now with Sigma Design Inc, Milpitus, CA 95035 USA (e-mail: wgzheng@ieee.org).

J. Luo was with the Computer Science and Engineering Department, University of Texas, Arlington, TX 96019 USA. He is now with Thomson Corporate Research, Princeton, NJ 08540 USA (e-mail: Daniel.luo@thomson.net).

M. Liou was with Hong Kong University of Science and Technology, Hong Kong, SAR. He is now with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, SAR.

Digital Object Identifier 10.1109/TCSVT.2006.870022

video sequences, and the reasons for that are not obvious. The second problem is their high complexity. MV prediction (MVP) techniques have shows significant performance improvements. An initial guess of the next MV is obtained by prediction from the previous MV in temporal or spatial domain. Various MVP are proposed in literatures [6], [14], [23], [30], [36]. Within the MPEG-4 or H.26X framework, MVs are differentially coded. The predicted MV is subtracted from the actual MV and the resulting difference is coded using a variable length code.

MPEG and ITU series standards recommend block-based motion compensation techniques. After considerable assessment and rigorous testing, the MPEG-4 Part-7 adopted MV field adaptive fast search technique (MVFAST) [8]–[10] as a recommended ME algorithm [13]. Our group proposed an algorithm named predictive MVFAST (PMVAST) [34] (part of which appeared in [13]) that is also recommended in the MPEG-4 standard. In this paper, we propose a ME algorithm that is a combination of a number of novel ideas for finding more accurate MVs and with a faster speed. The proposed algorithm, named as fast adaptive ME (FAME) algorithm, outperforms MVFAST and PMVFAST. FAME takes advantages of the correlation between MVs in both spatial and temporal domains, controls to curb the search, avoids search stationary regions, and adaptively uses special diamond shape search patterns to accelerate motion search. The algorithm yields very close quality compared to the full search but with several hundred times speedup.

The rest of this paper is organized as follows. Section II provides an overview of MVFAST and PMVFAST. The details and descriptions of these algorithms set the stage for the discussion of FAME that uses new features but takes into account some of the good features of these two previous algorithms. Section III describes the proposed algorithm in details. Section IV provides extensive performance evaluation and comparisons, as well as the comments and observations. The last section concludes the paper with final remarks and future directions.

II. Overview of MVFAST and PMVFAST

MVFAST offers high performance both in quality and speed and does not require memory to store the searched points and MVs. The MVFAST has been adopted by MPEG-4 Part 7 (March 2000) as the core technology for fast ME.

MVFAST exploits an optional phase called early elimination of search as its first step. In the phase of early elimination of search, the search for a MB will be terminated immediately, if its SAD value obtained at (0, 0) is less than a threshold T, and the MV is assigned as (0, 0). MVFAST defines local motion activity of a MB to classify motion types as high, medium or low. The local motion activity determines the initial search center as well as the search strategy. If the motion activity is low or medium, the search center is the origin. Otherwise, the vector belonging to the set of MVs in the region of support (ROS) that yields the minimum SAD is chosen as the search center.

MVFAST employs two search patterns to perform local search around the search center: the small diamond search pattern (SDSP) and large diamond search pattern (LDSP). The choice depends on the motion activity identified. If the motion activity is low or high, SDSP is employed; otherwise, LDSP is chosen. LDSP switches to SDSP, if the center position gives the minimum SAD.

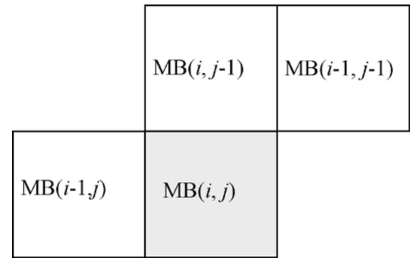

Fig. 1. BMVCL.

A derivative of MVFAST, called PMVFAST is considered as an optional approach that might benefit in special coding situations. PMVFAST incorporates a set of thresholds in MVFAST to trade higher speed-up at the cost of memory size, memory bandwidth and additional algorithmic complexity. PMVFAST combines the ‘stop when good enough’ principle, the threshold of stopping criteria and the spatial and temporal MV prediction of advanced predictive diamond zonal search (APDZS) [33], [35] as well as the efficient LDSP and SDSP of MVFAST. The PMV is used as the initial predictor. The search stops if the PMV satisfies the stopping criterion. PMVFAST computes the SAD of some highly probable MVs and stops if the minimum SAD so far satisfied the stopping criterion. PMVFAST performs a local search using some of the techniques of MVFAST.

III. FAME Algorithm

Motion is generally classified as foreground and background motion. Most ME algorithms assume the background is still or has slow motion, and that the foreground motion is stable (moving in constant direction). This assumption leads to a certain correlation between MVs, which are then searched using a fixed shape, such as a diamond or square; MVFAST and PMVFAST use similar ideas. However, the above assumptions are not always true. In many cases the foreground is almost still but the background moves fast, such as the camera focusing on a moving car. Here, the car may be relatively still but the background may have fast motion. In such situations, a fixed search pattern may get trapped in a local minimal, leading to incorrect motion prediction. Further, a fixed threshold cannot adapt to different kinds of sequence, and, wastes useful computational resource. The FAME algorithm takes advantages of the correlation between MVs in both spatial and temporal domains, and uses adaptive shape search patterns to accelerate motion search. Various features of FAME are described next.

Adaptive Threshold for Identifying Stationary Blocks: About 98% of the stationary blocks have their SAD at position (0, 0) less than 512 for MB size of 16 × 16 [8], [13]. If we can detect a stationary MB, we can just set its MV as (0, 0) and skip the motion search. Previous algorithms [8], [34] detect stationary blocks using a fixed threshold. An adaptive threshold makes detection fast and more robust. Most stationary MBs have small SADs at (0, 0). The proposed algorithm uses an adaptive threshold, named threshold for stationary block (TSB), which

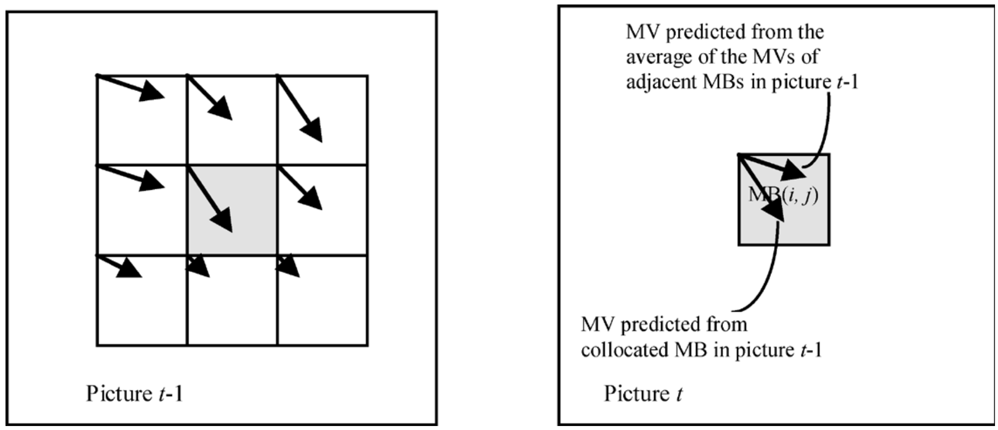

Fig. 2. Temporal MV prediction.

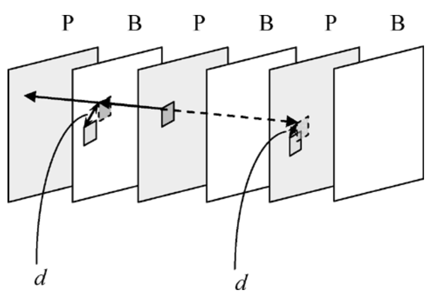

Fig. 3. Illustration of motion inertia.

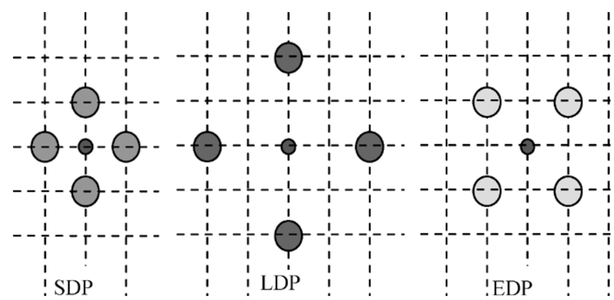

Fig. 4. Various search patterns used in FAME.

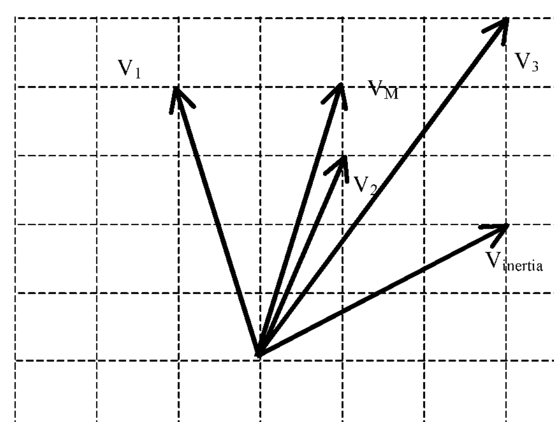

Fig. 5. Example of MV candidate list.

makes the detection faster and robust in the sense that it can resist the influence of noise. If the SAD at (0, 0) is less than TSB, the algorithm skips the rest of the search and use (0, 0) as the MV of the current block. TSB is determined as follows:

- denote by MVCL the MV candidate list. Initially, MVCL contains MVs of the upper, upper-right and left MBs.
- if all adjacent MBs in MVCL have MVs at (0, 0), the algorithm uses the maximum of their SAD as threshold TSB;
- if one of adjacent MBs in MVCL has MV unequal to (0, 0), we use the minimum of SAD of adjacent MBs as TSB since the possibility of current block inside stationary area become lower.

To make the detection more robust, we bound TSB within a certain range. The upper boundary can be tuned in terms of the type of the video sequence.

Detection of MBs Belonging to Same Moving Object: The smoothness of the motion field, especially within the same moving object, means high correlation of MVs. This property can be used to accelerate motion search if we can identify the current MB is within the same moving object of its adjacent MBs. This property is explained as follows:

$$l_i = |x_i - \overline{x}| + |y_i - \overline{y}| \quad (1)$$

$$L = Max\{l_i\} \quad (2)$$

where L is local motion activity (LMA) measurement factor, $(\overline{x}, \overline{y})$ is the average of MVs of the upper, upper-right, and left MBs, and (x_i, y_i) is the ith MV in MVCL.

The definition of LMA in FAME is different from that in MVFAST. In MVFAST, l_i is defined as the cityblock length of MV_i [8], [13]. In FAME, l_i is defined as the cityblock length of the variation of MV. The notion of our LMA makes the measurement more localized and more detailed. It also reflects the consistency of MVs within MVCL. It can be observed that if the current MB and MBs in MVCL belong to the same moving object, L should have a small value. The LMA is defined as

$$\text{LMA} = \begin{cases} \text{Low}, & \text{if } L \le L_1 \\ \text{Medium}, & \text{if } L_1 < L \le L_2 \\ \text{High}, & \text{if } L > L_2. \end{cases} \quad (3)$$

In our experiments, $L_1 = 1$, $L_2 = 4$. When LMA is low, it means its adjacent MBs have similar motion property. The current MB may possibly be inside the same object with its adjacent MBs, and may have the same MV.

Adaptive Threshold to Enable Half-Stop: In order to avoid being trapped in unnecessary search, FAME uses a threshold to

TABLE I
ABBREVIATIONS USED IN SUBSEQUENT TABLES

Abbreviations	Explanation	Abbreviations	Explanation
FS	Full Search motion estimation algorithm	MVF	Motion Vector Field Adaptive Search Technique
PMV	Predictive Motion Vector Field Adaptive Search Technique	FAME	Fast Adaptive Motion Estimation
BR	Bit Rate, the unit is kbits per second.	FR	Frame Rate, the unit is frames per second
YPSNR	Peak Signal-to-noise Ratio for Y Component; the unit is DB	UPSNR	Peak Signal-to-noise Ratio for U Component; the unit is DB
VPSNR	Peak Signal-to-noise Ratio for V Component; the unit is DB	ChkPts	Number of Checkpoints
SR	Search Range, the unit is pixels. The search window size is (2SR+1) by (2SR+1)	Search speed increased	The percentage of the search time saved by FAME over the designate motion estimation algorithm. It is obtained by (Search time of the designate algorithm – search time of FAME) / search time of FAME.
ST-sp	Speedup according to the search time. This value is obtained by divide the search time of the full search algorithm by the search time of the designate motion estimation algorithm.	PSNRgain	Peak Signal-to-noise Rate Gain. This value is obtained by subtract YPSNR value of the full search algorithm from the YPSNR value of the designate motion estimation algorithm.
TSBUB	Upper bound of TSB	THSUB	Upper bound of THS
Search Time	Search Time is the time spent on the motion searh. The time unit of search time is clock tick. The frequency of clock tick depends on the computer. In our experiments, the frequency of clock tick is 3579545 ticks per second.	ChPt-sp	Speedup according to the number of checkpoints. This value is obtained by dividing the number of checkpoints in the full search algorithm by the number of checkpoints in the designate motion estimation algorithm.
Chkpts saved	The percentage of checkpoints saved by FAME over the designate motion estimation algorithm. It is obtained by (Chkpts of the designate algorithm – Chkpts of FAME) / Chkpts of FAME.	LSAD-sp	Speedup according to the number of lines of SAD. This value is obtained by dividing the Line-SAD value of the full search algorithm by the Line-SAD value of the designate motion estimation algorithm.
Line-SAD	Number of Lines of SAD. A block SAD calculation is composed of lines of SADs. Half stop technique is used in block SAD calculations. After the computation of a line's SAD, the line SAD value is added to the partial SAD of that block. If the partial SAD value is larger than the minimum block SAD so far, it is not necessary to compute the SAD for the rest of the lines in the block. This technique can save computation without losing accuracy.	Line-SAD saved	The percentage of lines of SAD saved by FAME over the designate motion estimation algorithm. It is obtained by (Line-SAD of the designate algorithm – Line-SAD of FAME) / Line-SAD of FAME.

stop the search when the result is good enough. If the prediction error during the search is below this threshold, the search stops earlier. We call this the threshold for half-stop (THS), which is again adaptively set according to LMA of its MVCL.

- THS is equal to the mean value of SAD of adjacent MBs if the local motion activity is less than 4.
- THS is equal to the minimum value of SAD of adjacent MB.
- To avoid the extremely low or high value of THS caused by noise, we bound THS within a certain range. Generally, the lower bound has the same value of the lower bound of TSB. The upper bound can be tuned in terms of the type of video sequence.

Extended MV Candidate Set: The MVCL initially consists of a basic MVCL, called BMVCL. This includes MVs from its neighbor MBs as shown in Fig. 1, where MB (i, j) is the current MB. The BMVCL is extended to include additional MVs, as explained below.

Since the FAME algorithm is also based on motion prediction technique, the candidates in MVCL are crucial for speed and quality of FAME. If there are more MV candidates, the chance to find true vector faster and more precise is higher. Fig. 2 shows ordinary members in MVCL, presenting the prediction from the spatial domain. However, the correlation of MVs does not exist only in the spatial domain, but exists in the temporal domain as well. FAME uses one MV prediction from the temporal domain.

Motion Inertia: Some temporal MV prediction based algorithms are reported [8], [14], [33]. But these algorithms use only the average of MVs of adjacent MBs in the reference picture, or the MV of corresponding MB in the reference picture. As shown in Fig. 2, for MB (i, j), the prediction of MV from temporal domain would be $V_{t-1}(i, j)$, or the average of MVs of adjacent MBs in picture $t-1$. Based on our experiments and observations, the motion track of a moving object in a video sequence is continuous except when scene change occurs. That means there is a so-called *motion inertia* property in the temporal domain. The property can be used for MV prediction and interpolation.

Let (x_0, y_0) present the coordinates of left-top corner of current MB. V_{inertia} denotes the inertia MV predictor of current MB. $\widetilde{V}$ is the set of MVs of the reference frame. Let (x_{t-1}, y_{t-1}) be the coordinates of the left-top corner of MB in the reference frame and $V_{t-1} \in \widetilde{V}$, $V_{t-1} = (v_{x,t-1}, v_{y,t-1})$ be the MV of this MB. Based on the inertia property (the MB moves with the same MV), the position of MB in current frame will be

$$x_t = x_{t-1} + v_{x,t-1} \tag{4}$$

$$y_t = y_{t-1} + v_{y,t-1}. \tag{5}$$

Let $d = |x_0 - x_t| + |y_0 - y_t|$. The goal of MV prediction is to find out a MV in the reference frame that minimizes d, as (6). Fig. 3 illustrates the inertia MVs.

$$V_{\text{inertia}} = \arg\min_{V} \sum d. \tag{6}$$

TABLE II
DETAILED COMPARISON OF FAME WITH FS, MVFAST, AND PMVFAST

Sequence	BR	FR	SR		YPSNR	UPSNR	VPSNR	Bits	ChkPts	Line-SAD	SearchTime	ChPt-sp	LSAD-sp	ST-sp	PSNRgain
Cheerleader (ccir 720X480)	4000	30	64	FS	29.425	31.765	32.650	20010966	9511788726	48963643590	52876521460	1.00	1.00	1.00	0.000
				MVF	29.220	31.822	32.725	20010646	4188979	57300927	68680412	2270.67	854.50	769.89	–0.205
				PMV	28.950	31.660	32.569	20004198	4929845	72536859	85357957	1929.43	675.02	619.47	–0.475
				FAME	29.307	31.853	32.759	20010406	3468725	48375530	67251127	2742.16	1012.16	786.25	–0.118
			128	FS	29.313	31.650	32.533	19744574	31055725116	1.50029E+11	1.67227E+11	1.00	1.00	1.00	0.000
				MVF	29.115	31.744	32.657	19745038	4199261	57489855	69534450	7395.52	2609.67	2404.94	–0.198
				PMV	28.852	31.574	32.493	19739582	4939382	72704360	85735712	6287.37	2063.56	1950.49	–0.461
				FAME	29.221	31.784	32.7	19747502	3475139	48474341	71155300	8936.54	3095.03	2350.16	–0.092
Cheerleader (ccir 720X480)	9000	30	64	FS	33.778	35.339	36.117	44402238	10040641104	47759902109	59044262168	1.00	1.00	1.00	0.000
				MVF	33.764	35.387	36.202	44398870	4138731	55053487	67066425	2426.02	867.52	880.38	–0.014
				PMV	33.624	35.280	36.093	44403782	4676206	69328258	81824476	2147.18	688.90	721.60	–0.154
				FAME	33.778	35.386	36.196	44402142	3436669	47503525	66261903	2921.62	1005.40	891.07	0.000
			128	FS	33.766	35.321	36.095	44400222	32756437918	1.45911E+11	1.63066E+11	1.00	1.00	1.00	0.000
				MVF	33.752	35.381	36.196	44398926	4142424	55112378	67556335	7907.55	2647.53	2413.77	–0.014
				PMV	33.615	35.269	36.088	44404126	4680034	69377110	82181903	6999.19	2103.16	1984.20	–0.151
				FAME	33.763	35.376	36.17	44402630	3441746	47566307	70024530	9517.39	3067.54	2328.69	–0.003
Flower Garden (ccir 720X480)	4000	30	64	FS	29.423	32.402	34.081	19768446	9841304792	41198077209	52724019817	1.00	1.00	1.00	0.000
				MVF	29.147	32.286	33.994	19773662	3228815	36480809	49896518	3047.96	1129.31	1056.67	–0.276
				PMV	25.201	30.428	32.463	19718078	4864019	68806582	87450378	2023.29	598.75	602.90	–4.222
				FAME	29.215	32.315	34.016	19767054	1602768	18948783	36535806	6140.19	2174.18	1443.08	–0.208
			128	FS	29.394	32.378	34.064	19769174	32139459256	1.21147E+11	1.09406E+11	1.00	1.00	1.00	0.000
				MVF	29.112	32.257	33.977	19772326	3230152	36515798	50212299	9949.83	3317.67	2178.87	–0.282
				PMV	25.184	30.419	32.464	19709718	4865892	68831709	87484387	6605.05	1760.05	1250.58	–4.210
				FAME	29.179	32.287	33.996	19766622	1604483	18975121	45044101	20031.04	6384.54	2428.87	–0.215
Flower Garden (ccir 720X480)	9000	30	64	FS	33.712	35.861	36.628	44434390	10162085820	38675482264	49555325099	1.00	1.00	1.00	0.000
				MVF	33.506	35.782	36.540	44433358	3181659	34080035	47253909	3193.96	1134.84	1048.70	–0.206
				PMV	30.423	33.490	34.865	44438150	4755679	64975558	84291698	2136.83	595.23	587.90	–3.289
				FAME	33.518	35.773	36.534	44432742	1546122	17801917	34551283	6572.63	2172.55	1434.25	–0.194
			128	FS	33.774	35.908	36.666	45034966	33153562664	1.12899E+11	1.01957E+11	1.00	1.00	1.00	0.000
				MVF	33.566	35.824	36.574	45034670	3181031	34055390	47071260	10422.27	3315.16	2166.02	–0.208
				PMV	30.491	33.541	34.900	45038630	4750945	64909010	84831355	6978.31	1739.34	1201.88	–3.283
				FAME	33.576	35.813	36.569	45034774	1546503	17802805	43152934	21437.76	6341.64	2362.70	–0.198
Coastguard (cif 352X288)	112	10	16	FS	27.905	41.422	42.765	1122198	138312976	1017346319	1136439206	1.00	1.00	1.00	0.000
				MVF	28.134	41.613	42.849	1102654	793482	10178636	13050421	174.31	99.95	87.08	0.229
				PMV	27.229	41.163	42.518	1168214	889162	12430280	15877400	155.55	81.84	71.58	–0.676
				FAME	28.126	41.615	42.847	1099478	602476	7859812	11713025	229.57	129.44	97.02	0.221
			32	FS	27.812	41.401	42.765	1120982	533386562	3473871113	3786876366	1.00	1.00	1.00	0.000

TABLE II (*Continued*)
DETAILED COMPARISON OF FAME WITH FS, MVFAST, AND PMVFAST

Sequence	BR	FR	SR		YPSNR	UPSNR	VPSNR	Bits	ChkPts	Line-SAD	SearchTime	ChPt-sp	LSAD-sp	ST-sp	PSNRgain
				MVF	28.371	41.742	42.928	1120582	783851	9895952	13101163	680.47	351.04	289.05	0.559
				PMV	27.219	41.150	42.518	1159822	890265	12446529	15841646	599.13	279.10	239.05	–0.593
				FAME	28.367	41.738	42.905	1120166	584464	7489203	11484824	912.61	463.85	329.73	0.555
Coastguard (qcif 176X144)	48	10	16	FS	30.746	42.495	42.996	479542	34846160	189599526	215061380	1.00	1.00	1.00	0.000
				MVF	30.749	42.465	42.970	479270	181941	1919485	2666013	191.52	98.78	80.67	0.003
				PMV	29.959	42.212	42.754	479750	169421	2145466	2901172	205.68	88.37	74.13	–0.787
				FAME	30.724	42.429	42.969	479414	106100	1185057	2050013	328.43	159.99	104.91	–0.022
			32	FS	30.683	42.442	42.974	479662	127065512	590207188	647641896	1.00	1.00	1.00	0.000
				MVF	30.710	42.436	42.951	479422	182478	1929129	2624308	696.33	305.94	246.79	0.027
				PMV	29.943	42.157	42.721	479830	169062	2138081	2955561	751.59	276.05	219.13	–0.740
				FAME	30.702	42.436	42.941	479350	106502	1190667	2016880	1193.08	495.69	321.11	0.019
Container (qcif 176X144)	10	7.5	16	FS	28.737	36.643	35.235	105606	12150768	54490416	59445510	1.00	1.00	1.00	0.000
				MVF	28.782	36.721	35.257	104886	27949	302344	429234	434.75	180.23	138.49	0.045
				PMV	28.798	36.727	35.322	104958	29678	363401	537740	409.42	149.95	110.55	0.061
				FAME	28.793	36.643	35.261	105694	23848	246806	424354	509.51	220.78	140.08	0.056
			32	FS	28.716	36.643	35.332	105374	41802498	153324275	161451675	1.00	1.00	1.00	0.000
				MVF	28.730	36.630	35.299	104854	27890	303197	429181	1498.83	505.69	376.19	0.014
				PMV	28.727	36.603	35.236	105430	30501	374890	592081	1370.53	408.98	272.69	0.011
				FAME	28.746	36.63	35.308	105134	21681	224107	382599	1928.07	684.16	421.99	0.030
Foreman (cif 352X288)	1024	30	16	FS	36.308	41.274	42.784	10248766	414565280	1966578957	2217412491	1.00	1.00	1.00	0.000
				MVF	36.270	41.258	42.788	10245926	2144869	26038220	34040603	193.28	75.53	65.14	–0.038
				PMV	34.717	40.235	41.694	10246046	2231551	32918196	41126347	185.77	59.74	53.92	–1.591
				FAME	36.194	41.199	42.735	10243222	1267151	15899751	24609629	327.16	123.69	90.10	–0.114
			32	FS	36.450	41.404	42.913	10247006	1597273676	5844138319	6417758637	1.00	1.00	1.00	0.000
				MVF	36.376	41.344	42.887	10244222	2147248	25984233	34304537	743.87	224.91	187.08	–0.074
				PMV	34.733	40.252	41.707	10242774	2258148	33319044	41779718	707.34	175.40	153.61	–1.717
				FAME	36.313	41.316	42.859	10240630	1263609	15665946	24578419	1264.06	373.05	261.11	–0.137
			48	FS	36.475	41.430	42.935	10246214	3422933360	10906541251	11842335317	1.00	1.00	1.00	0.000
				MVF	36.384	41.361	42.904	10245926	2138988	25825944	34466406	1600.26	422.31	343.59	–0.091
				PMV	34.718	40.243	41.693	10247614	2261853	33370815	41632553	1513.33	326.83	284.45	–1.757
				FAME	36.32	41.317	42.862	10243230	1256764	15525367	25130325	2723.61	702.50	471.24	–0.155
Foreman (cif 352X288)	112	10	16	FS	31.538	38.376	39.612	1123102	131388144	694618502	760185705	1.00	1.00	1.00	0.000
				MVF	32.440	38.969	40.454	1111046	732698	9438788	12171120	179.32	73.59	62.46	0.902
				PMV	31.298	38.510	39.762	1110454	619136	9169132	11730459	212.21	75.76	64.80	–0.240
				FAME	32.51	39.024	40.557	1110726	460604	5917730	9077969	285.25	117.38	83.74	0.972
			32	FS	31.403	38.294	39.361	1125086	507311762	2138638952	2282294416	1.00	1.00	1.00	0.000
				MVF	32.482	38.961	40.481	1111918	737816	9497643	12020203	687.59	225.18	189.87	1.079
				PMV	31.303	38.522	39.790	1110990	623252	9233316	11766649	813.98	231.62	193.96	–0.100

TABLE II (*Continued*)
DETAILED COMPARISON OF FAME WITH FS, MVFAST, AND PMVFAST

Sequence	BR	FR	SR		YPSNR	UPSNR	VPSNR	Bits	ChkPts	Line-SAD	SearchTime	ChPt-sp	LSAD-sp	ST-sp	PSNRgain
				FAME	32.553	39.039	40.607	1112966	464313	5936919	9148308	1092.61	360.23	249.48	1.150
Foreman (cif 352X288)	512	15	16	FS	37.416	42.154	44.099	5119726	204431040	863767043	993550327	1.00	1.00	1.00	0.000
				MVF	37.337	42.093	44.068	5120470	991931	11577270	15472443	206.09	74.61	64.21	–0.079
				PMV	36.233	41.392	43.355	5121654	831884	12174599	16306278	245.74	70.95	60.93	–1.183
				FAME	37.245	42.016	44.008	5120622	539861	6631956	10434967	378.67	130.24	95.21	–0.171
			32	FS	37.399	42.153	44.082	5120566	787584890	2634528574	2927037395	1.00	1.00	1.00	0.000
				MVF	37.338	42.098	44.064	5120718	998213	11668611	15673484	788.99	225.78	186.75	–0.061
				PMV	36.232	41.399	43.360	5121302	834455	12212881	16295035	943.83	215.72	179.63	–1.167
				FAME	37.254	42.065	44.051	5120246	545025	6672340	10815112	1445.04	394.84	270.64	–0.145
			48	FS	37.374	42.137	44.069	5120022	1687209488	4924327380	5389924601	1.00	1.00	1.00	0.000
				MVF	37.307	42.085	44.041	5120046	998665	11674712	15681968	1689.46	421.79	343.70	–0.067
				PMV	36.198	41.377	43.334	5122694	833784	12206887	16276819	2023.56	403.41	331.14	–1.176
				FAME	37.218	42.014	43.985	5120198	547550	6706190	11256641	3081.38	734.30	478.82	–0.156
Hall Monitor (qcif 176X144)	10	7.5	16	FS	28.390	35.658	39.152	104718	10465408	42391423	46127398	1.00	1.00	1.00	0.000
				MVF	28.365	35.633	39.093	103806	32917	387739	570749	317.93	109.33	80.82	–0.025
				PMV	28.404	35.640	39.120	103142	26955	373360	573385	388.25	113.54	80.45	0.014
				FAME	28.416	35.649	39.117	103422	23169	265897	426171	451.70	159.43	108.24	0.026
			32	FS	28.381	35.669	39.171	105222	39321366	138293935	146255894	1.00	1.00	1.00	0.000
				MVF	28.315	35.589	39.094	103374	32355	383581	530140	1215.31	360.53	275.88	–0.066
				PMV	28.409	35.639	39.102	103398	27512	378239	545264	1429.24	365.63	268.23	0.028
				FAME	28.358	35.604	39.093	103286	22952	263362	467726	1713.20	525.11	312.70	–0.023
Mother & Daughter (qcif 176X144)	24	10	16	FS	35.358	41.112	41.449	240758	24412096	91688196	103516928	1.00	1.00	1.00	0.000
				MVF	35.266	41.021	41.319	240734	61033	687763	1037964	399.98	133.31	99.73	–0.092
				PMV	35.135	40.906	41.250	240846	47646	727661	1179810	512.36	126.00	87.74	–0.223
				FAME	35.298	41.02	41.354	240798	41767	485997	909203	584.48	188.66	113.85	–0.060
			32	FS	35.338	41.074	41.350	240710	92145284	278746673	300511057	1.00	1.00	1.00	0.000
				MVF	35.265	40.981	41.282	240718	60833	683861	1025679	1514.73	407.61	292.99	–0.073
				PMV	35.082	40.897	41.221	240486	47768	731996	1107156	1929.02	380.80	271.43	–0.256
				FAME	35.174	40.892	41.265	240710	42742	499106	951086	2155.85	558.49	315.97	–0.164
News (cif 352X288)	112	15	16	FS	33.254	38.741	39.563	1120614	109819168	462852114	515442881	1.00	1.00	1.00	0.000
				MVF	33.242	38.803	39.591	1119958	353587	4109849	5668365	310.59	112.62	90.93	–0.012
				PMV	33.157	38.715	39.475	1120502	264578	3773725	5360279	415.07	122.65	96.16	–0.097
				FAME	33.307	38.88	39.654	1119878	244864	2817341	4557334	448.49	164.29	113.10	0.053
			32	FS	33.249	38.737	39.542	1119718	427025996	1416108820	1529769426	1.00	1.00	1.00	0.000
				MVF	33.224	38.805	39.583	1119950	353995	4125401	5658913	1206.31	343.27	270.33	–0.025
				PMV	33.063	38.650	39.414	1119878	266310	3801828	5500532	1603.49	372.48	278.11	–0.186
				FAME	33.3	38.853	39.629	1120078	245999	2832111	4708318	1735.89	500.02	324.91	0.051
News (cif 352X288)	48	7.5	16	FS	32.382	37.463	38.562	498942	53663024	229099430	250364111	1.00	1.00	1.00	0.000

TABLE II (*Continued*)
DETAILED COMPARISON OF FAME WITH FS, MVFAST, AND PMVFAST

Sequence	BR	FR	SR		YPSNR	UPSNR	VPSNR	Bits	ChkPts	Line-SAD	SearchTime	ChPt-sp	LSAD-sp	ST-sp	PSNRgain
				MVF	32.536	37.639	38.708	498822	154363	1770846	2402253	347.64	129.37	104.22	0.154
				PMV	32.404	37.512	38.614	498502	112378	1595651	2388670	477.52	143.58	104.81	0.022
				FAME	32.49	37.556	38.634	498030	109331	1244951	2103255	490.83	184.02	119.04	0.108
			32	FS	32.391	37.441	38.553	498662	208286090	713018729	754363336	1.00	1.00	1.00	0.000
				MVF	32.560	37.657	38.707	499934	152879	1759159	2403431	1362.42	405.32	313.87	0.169
				PMV	32.318	37.434	38.541	497878	115229	1634235	2358699	1807.58	436.30	319.82	–0.073
				FAME	32.521	37.636	38.718	499478	105324	1203413	2068072	1977.57	592.50	364.77	0.130
Silence (qcif 176X144)	24	10	16	FS	30.933	35.520	37.161	240742	24735904	101465319	113694664	1.00	1.00	1.00	0.000
				MVF	30.934	35.539	37.168	240478	88023	1067370	1435090	281.02	95.06	79.22	0.001
				PMV	30.842	35.467	37.108	240430	88052	1204274	1638650	280.92	84.25	69.38	–0.091
				FAME	30.974	35.576	37.21	240822	69530	837233	1332698	355.76	121.19	85.31	0.041
			32	FS	30.913	35.497	37.133	241214	91959056	311014220	337945663	1.00	1.00	1.00	0.000
				MVF	30.880	35.474	37.122	240766	88548	1076258	1501308	1038.52	288.98	225.10	–0.033
				PMV	30.836	35.475	37.138	240678	87639	1198234	1679425	1049.29	259.56	201.23	–0.077
				FAME	30.944	35.536	37.173	240950	70825	853727	1343523	1298.40	364.30	251.54	0.031
									Average		FS	1.00	1.00	1.00	0.000
											MVF over FS	2011.67	673.34	544.17	0.033
											PMV over FS	1723.03	483.52	406.44	–0.901
											FAME over FS	3287.83	1053.16	610.93	0.040

For bidirectionally predicted frames, the MVs can be interpolated from its reference pictures. The interpolated MVs are computed by inertia criteria and scaling. The MVs are used as $V_{\rm inertia}$ for B-frames. From observation, we found that the interpolated MV can provide good prediction of MV for B-frames, and this the motion search can be terminated quickly.

When LMA is not low, $V_{\rm inertia}$ and an additional MV V_M resulting from the mean filter is added to the BMVCL. $V_M = (V_{Mx}, V_{My})$, where V_{Mx} and V_{My} are obtained from (7)

$$\begin{aligned} V_{Mx} &= \text{Mean}(V_{1x}, V_{2x}, V_{3x}) \\ V_{My} &= \text{Mean}(V_{1y}, V_{2y}, V_{3y}) \end{aligned} \tag{7}$$

where V_1, V_2, and V_3 are the MVs of the upper, upper-right, and left MBs.

For instance, if $V_1 = (-1, 4)$, $V_2 = (1, 3)$ and $V_3 = (3, 5)$, then $V_{Mx} = 1$ and $V_{My} = 4$.

Adaptive Search Patterns: FAME adaptively uses four motion search patters: small diamond pattern (SDP), large diamond pattern (LDP), elastic diamond pattern (EDP), and motion adaptive pattern (MAP). The first three patterns are shown as Fig. 4. LDP and EDP are used to locate raw MV inside search range, while SDP is used to fine tune an MV. MAP is used to find predictors in the MVCL. The algorithm selects these patterns adaptively depending upon LMA. The algorithm switches these patterns when needed.

The search patterns are selected according to the following principles.

- SDP is used to refine a predicted MV. Once the minimum of SAD is located at the center of diamond, the center represents the MV, and the search can be terminated.
- If the number of successively executed SDPs exceeds a certain predefined constant, *called elastic factor* K, the search pattern switches to EDP.
- EDP and LDP are used for fast wide-range search in diagonal direction and in horizontal and vertical direction respectively and to avoid the search from being trapped at local minima. The algorithm switch to LDP once after executing EDP. LDP will switch to EDP when the minimum of SAD is not located at the center. Other wise, it will switch to SDP for further refinement.
- MAP is used to search the predictors in MVCL. It is used first if the *local motion activity* is not low.

Search Strategy: The search strategy is as follows.

1) Add the spatial predictors to BMVCL, as shown in Fig. 5. Compute LMA with (1)–(3).
2) When $\text{LMA} = \text{Low}$, the search starts from SDP, and elastic factor $K = 2$, and initial search center is the average in MVCL.

TABLE III
SUMMARY OF PERFORMANCE COMPARISON WITH MVFAST, PMVFAST

Sequence	BR	FR	SR	Chkpts saved over		Line-SAD saved over		Search speed faster than		PSNR Gain over	
				MVF	PMV	MVF	PMV	MVF	PMV	MVF	PMV
Cheerleader (ccir 720X480)	4000	30	64	20.76%	42.12%	18.45%	49.95%	2.13%	26.92%	0.087	0.357
			128	20.84%	42.13%	18.60%	49.99%	-2.28%	20.49%	0.106	0.369
Cheerleader (ccir 720X480)	9000	30	64	20.43%	36.07%	15.89%	45.94%	1.21%	23.49%	0.014	0.154
			128	20.36%	35.98%	15.86%	45.85%	-3.52%	17.36%	0.011	0.148
Flower Garden (ccir 720X480)	4000	30	64	101.45%	203.48%	92.52%	263.12%	36.57%	139.36%	0.068	4.014
			128	101.32%	203.27%	92.44%	262.75%	11.47%	94.22%	0.067	3.995
Flower Garden (ccir 720X480)	9000	30	64	105.78%	207.59%	91.44%	264.99%	36.76%	143.96%	0.012	3.095
			128	105.69%	207.21%	91.29%	264.60%	9.08%	96.58%	0.010	3.085
Coastguard (cif 352X288)	112	10	16	31.70%	47.58%	29.50%	58.15%	11.42%	35.55%	-0.008	0.897
			32	34.11%	52.32%	32.14%	66.19%	14.07%	37.94%	-0.004	1.148
Coastguard (qcif 176X144)	48	10	16	71.48%	59.68%	61.97%	81.04%	30.05%	41.52%	-0.025	0.765
			32	71.34%	58.74%	62.02%	79.57%	30.12%	46.54%	-0.008	0.759
Container (qcif 176X144)	10	7.5	16	17.20%	24.45%	22.50%	47.24%	1.15%	26.72%	0.011	-0.005
			32	28.64%	40.68%	35.29%	67.28%	12.18%	54.75%	0.016	0.019
Foreman (cif 352X288)	1024	30	16	69.27%	76.11%	63.76%	107.04%	38.32%	67.11%	-0.076	1.477
			32	69.93%	78.71%	65.86%	112.68%	39.57%	69.99%	-0.063	1.580
			48	70.20%	79.97%	66.35%	114.94%	37.15%	65.67%	-0.064	1.602
Foreman (cif 352X288)	112	10	16	59.07%	34.42%	59.50%	54.94%	34.07%	29.22%	0.070	1.212
			32	58.90%	34.23%	59.98%	55.52%	31.39%	28.62%	0.071	1.250
Foreman (cif 352X288)	512	15	16	83.74%	54.09%	74.57%	83.57%	48.27%	56.27%	-0.092	1.012
			32	83.15%	53.10%	74.88%	83.04%	44.92%	50.67%	-0.084	1.022
			48	82.39%	52.28%	74.09%	82.02%	39.31%	44.60%	-0.089	1.020
Hall Monitor (qcif 176X144)	10	7.5	16	42.07%	16.34%	45.82%	40.42%	33.92%	34.54%	0.051	0.012
			32	40.97%	19.87%	45.65%	43.62%	13.34%	16.58%	0.043	-0.051
Mother & Daughter (qcif 176X144)	24	10	16	46.13%	14.08%	41.52%	49.73%	14.16%	29.76%	0.032	0.163
			32	42.33%	11.76%	37.02%	46.66%	7.84%	16.41%	-0.091	0.092
News (qcif 176X144)	112	15	16	44.40%	8.05%	45.88%	33.95%	24.38%	17.62%	0.065	0.150
			32	43.90%	8.26%	45.67%	34.24%	20.19%	16.83%	0.076	0.237
News (cif 352X288)	48	7.5	16	41.19%	2.79%	42.24%	28.17%	14.22%	13.57%	-0.046	0.086
			32	45.15%	9.40%	46.18%	35.80%	16.22%	14.05%	-0.039	0.203
Silence (qcif 176X144)	24	10	16	26.60%	26.64%	27.49%	43.84%	7.68%	22.96%	0.040	0.132
			32	25.02%	23.74%	26.07%	40.35%	11.74%	25.00%	0.064	0.108
Average				53.92%	58.29%	50.70%	85.54%	20.85%	44.53%	0.01	0.94

3) When $\mathrm{LMA} = \mathrm{Median}$, MVCL is extended by adding V_M and V_{inertia}. The initial search pattern is the MAP. MVs in MVCL are checked respectively. The candidate with the minimum SAD is selected as the search center for subsequent search. Then the search pattern is SDP, and elastic factor $K = 4$.

4) When $\mathrm{LMA} = \mathrm{High}$, MVCL is extended by adding V_M and V_{inertia}. The initial pattern is MAP. The candidates in MVCL are checked respectively. The one with the minimum SAD is selected as new search center. Subsequent search is starts from SDP, and elastic factor $K = 1$.

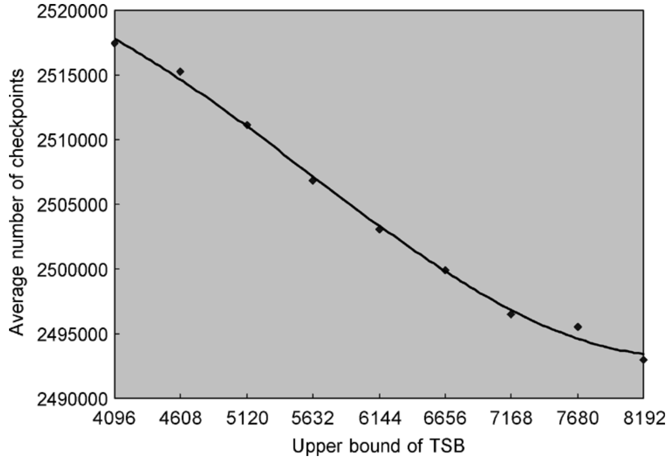

Fig. 6. Average number of checkpoints versus upper bound of TSB (CCIR format video).

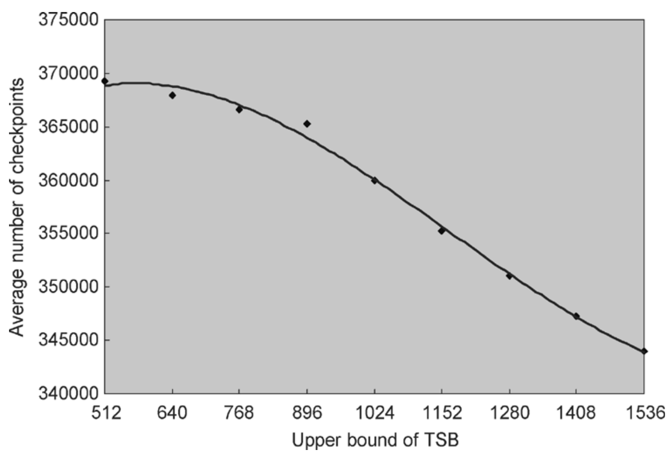

Fig. 7. Average number of checkpoints versus upper bound of TSB (QCIF and CIF format video).

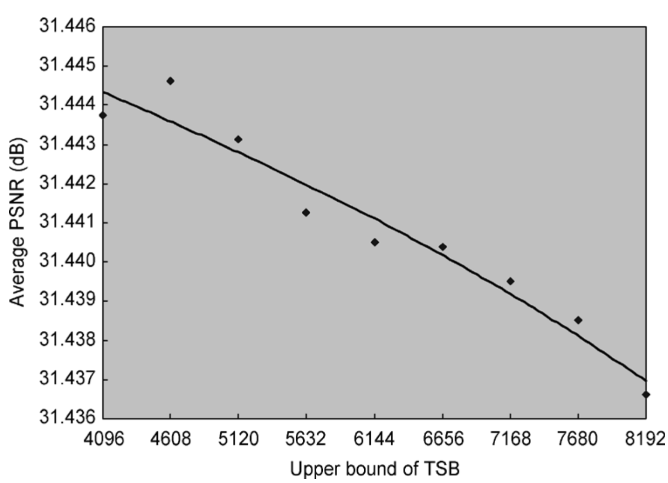

Fig. 8. Average PSNR versus upper bound of TSB (CCIR format video).

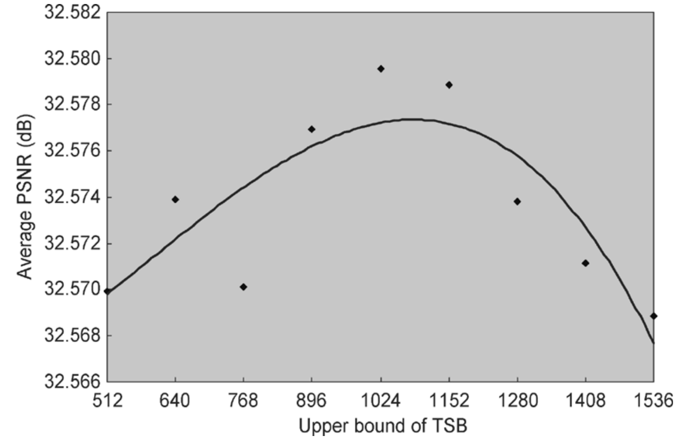

Fig. 9. Average PSNR versus upper bound of TSB (QCIF and CIF format video).

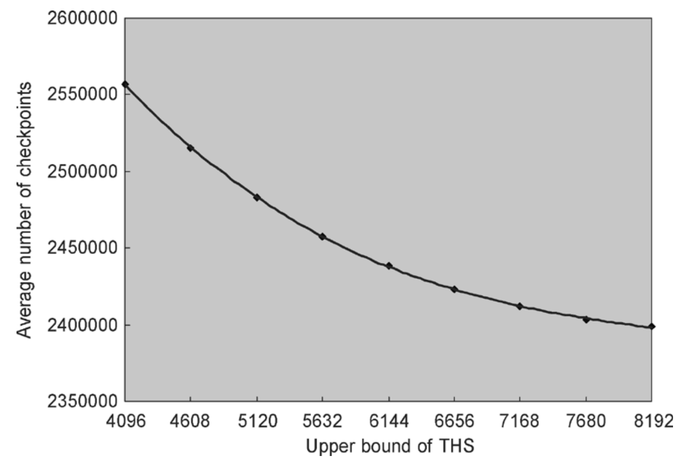

Fig. 10. Average number of checkpoints versus upper bound of THS (CCIR format video).

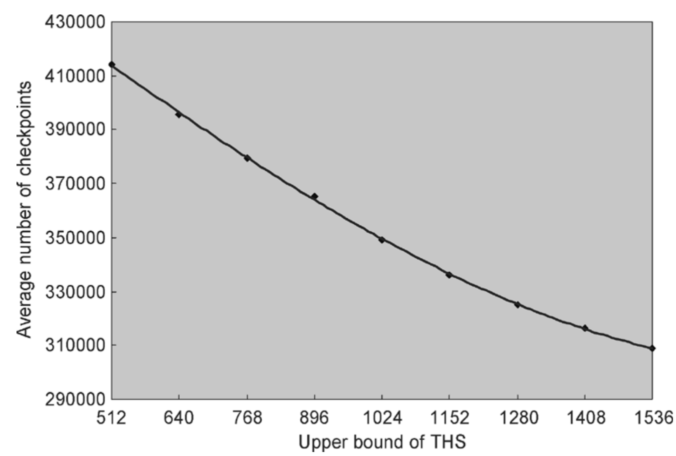

Fig. 11. Average PSNR versus upper bound of THS (CCIR format video).

Search Track: FAME keeps track of accessed checking points so as to avoid duplicate checking.

IV. Performance Evaluation

This section includes the performance of the proposed algorithm and its comparison with MVFAST and PMVFAST. Table I lists the abbreviations used in the comparison. Compared with these two supposedly the "best" algorithms, FAME is in faster and yields better visual quality, as demonstrated by the results presented in this section. The full search (FS) technique is also included for comparison. All ME algorithms being compared were implemented on Microsoft VM software. The rate control algorithm was TM5. The selection of the video sequences was according to MPEG-4 testing [15]. MVFAST and PMVFAST were implemented according to their description reported in

TABLE IV
PERFORMANCE COMPARISON WITH DIFFERENT UPPER BOUND OF TSB (CCIR FORMAT VIDEO)

Sequence	BR	FR	SR	TSBUB =4096		TSBUB = 4608		TSBUB =5120		TSBUB = 5632		TSBUB = 6144		TSBUB= 6656		TSBUB = 7168		TSBUB = 7680		TSBUB = 8192	
				YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts
Cheerleader (ccir 720X480)	4000	30	64	29.309	3470714	29.307	3468725	29.311	3466475	29.309	3452796	29.309	3448723	29.305	3442811	29.308	3437470	29.306	3433760	29.306	3433366
			128	29.217	3480094	29.221	3475139	29.219	3463995	29.219	3460707	29.213	3452438	29.221	3447852	29.218	3439730	29.217	3441074	29.217	3431715
Cheerleader (ccir 720X480)	9000	30	64	33.775	3441346	33.778	3436669	33.773	3427313	33.778	3420234	33.773	3413892	33.777	3409404	33.777	3403335	33.778	3400195	33.776	3397320
			128	33.763	3445266	33.763	3441746	33.76	3431886	33.758	3424157	33.759	3416913	33.764	3410552	33.763	3406256	33.764	3401836	33.762	3395231
Flower Garden (ccir 720X480)	4000	30	64	29.211	1603659	29.215	1602768	29.214	1602647	29.205	1601607	29.207	1600816	29.205	1600895	29.202	1598328	29.199	1598947	29.198	1599159
			128	29.18	1604495	29.179	1604483	29.179	1604423	29.178	1603574	29.177	1602605	29.169	1600916	29.17	1600890	29.169	1602181	29.163	1600115
Flower Garden (ccir 720X480)	9000	30	64	33.517	1547390	33.518	1546122	33.515	1546796	33.512	1546721	33.515	1544550	33.513	1543586	33.51	1543517	33.508	1543567	33.505	1544112
			128	33.578	1546585	33.576	1546503	33.574	1545547	33.571	1545071	33.571	1544543	33.569	1543483	33.568	1542138	33.567	1542539	33.566	1543165
Average				31.444	2517444	31.445	2515269	31.443	2511135	31.441	2506858	31.441	2503060	31.440	2499937	31.440	2496458	31.439	2495512	31.437	2493023

[13]. For the experiments reported in Tables II and III, both the upper bound of TSB and THS were 4608 for the CCIR format video sequences, and 896 for the QCIF and CIF format video sequences.

To investigate the effect of the upper bounds of TSB and THS, we examined a number of values. For the CCIR format video, we set the upper bounds of TSB and THS from 4K (4096) to 8K (8192), with steps of 512. For the QCIF and CIF format video, we set the upper bounds from 512 to 1536, with steps of 128. Tables VIII and IX show the effect of different values for L_1 and L_2.

A. Detailed Assessment and Comparison

We compare FAME with FS, MVFAST and PMVFAST, in terms of speed and quality. The number of checkpoints is the measure for search speed. From the results in the Table II, we observe that FAME has a speedup of 3000 over FS, and far outperforms MVFAST and PMVFAST in the terms of the number of checkpoints. The search time is another speed measure that takes into account the overhead of the algorithm. The overhead of the algorithm includes the time spent on setting the list, storing and fetching temporal and spatial MV candidates and computing motion inertia, etc. Table II indicates that FAME outperforms MVFAST and PMVFAST in the search time comparison as well. As for the visual quality comparison, MVFAST and FAME both slightly outperform FS, and are better than PMVFAST. FAME yields the best peak signal-to-noise ratio (PSNR) among the four algorithms.

B. Comparison Summary

Table III provides a summary of the performance comparison of FAME with MVFAST and PMVFAST. The percentages of checkpoints less than the reference algorithms are included in the table. On average, FAME is 54% faster than MVFAST and 58% faster than PMVFAST. In terms of PSNR values, FAME is slightly better than MVFAST by 0.01 dB and outperforms PMVFAST by almost 1 dB. We observe that FAME achieves

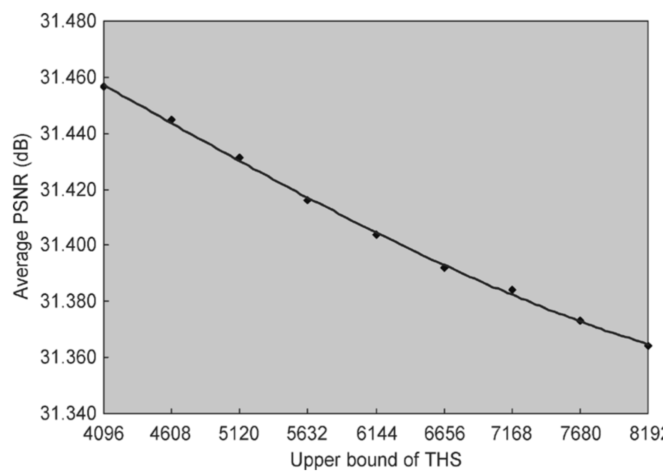

Fig. 12. Average number of checkpoints versus the upper bound of THS (QCIF and CIF format video).

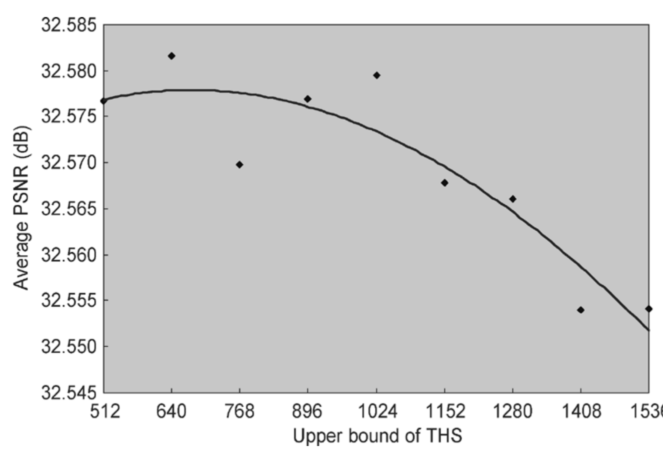

Fig. 13. Average PSNR versus the upper bound of THS (QCIF and CIF format video).

different speedup compared to MVFAST and PMVFAST on various video sequences.

Generally, motion in a scene is of two types: camera panning dominant or foreground object dominant. In the former,

TABLE V
PERFORMANCE OF FAME WITH DIFFERENT UPPER BOUND OF TSB (QCIF AND CIF FORMAT VIDEO)

Sequence	BR	FR	SR	TSBUB= 512		TSBUB= 640		TSBUB= 768		TSBUB = 896		TSBUB= 1024		TSBUB= 1152		TSBUB= 1280		TSBUB = 1408		TSBUB = 1536	
				YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts
Coastguard (cif 352X288)	112	10	16	28.115	604127	28.121	604975	28.126	604012	28.126	602476	28.127	597272	28.13	589555	28.134	579830	28.13	572264	28.135	563114
			32	28.349	583830	28.359	584330	28.354	584779	28.367	584464	28.366	577840	28.374	568481	28.374	560024	28.377	551066	28.371	543300
Coastguard (qcif 176X144)	48	10	16	30.684	106673	30.713	106505	30.726	106014	30.724	106100	30.74	102042	30.692	98660	30.719	94427	30.716	91874	30.685	89658
			32	30.664	106856	30.665	106808	30.716	106677	30.702	106502	30.666	103045	30.746	98641	30.703	95281	30.709	92159	30.684	89580
Container (qcif 176X144)	10	7.5	16	28.737	23580	28.799	24591	28.797	24360	28.793	23848	28.762	21118	28.759	20870	28.723	20841	28.723	20387	28.757	19392
			32	28.763	22761	28.747	22214	28.747	21958	28.746	21681	28.73	22587	28.73	22206	28.724	20590	28.724	20157	28.718	19682
Foreman (cif 352X288)	1024	30	16	36.202	1280206	36.202	1275128	36.203	1270585	36.194	1267151	36.197	1257396	36.19	1247363	36.189	1239637	36.196	1231173	36.186	1224924
			32	36.315	1273244	36.321	1267205	36.312	1265255	36.313	1263609	36.309	1251820	36.303	1241980	36.305	1232752	36.307	1225225	36.303	1217607
			48	36.332	1267399	36.328	1262304	36.323	1257844	36.32	1256764	36.312	1244233	36.311	1235581	36.315	1225636	36.31	1218480	36.319	1211016
Foreman (cif 352X288)	112	10	16	32.51	464598	32.505	464161	32.51	462769	32.51	460604	32.523	453189	32.517	447240	32.502	443415	32.503	438181	32.497	435740
			32	32.535	468353	32.531	467717	32.543	464817	32.553	464313	32.544	457413	32.548	449247	32.551	446114	32.546	441600	32.559	438827
Foreman (cif 352X288)	512	15	16	37.271	545809	37.252	543846	37.249	541586	37.245	539861	37.237	533522	37.231	529558	37.235	525088	37.234	523151	37.223	521840
			32	37.269	551591	37.257	548887	37.256	547373	37.254	545025	37.236	539926	37.24	534831	37.241	531541	37.236	528263	37.237	526356
			48	37.238	553825	37.231	550774	37.225	548952	37.218	547550	37.217	540577	37.204	536714	37.196	533475	37.202	529821	37.202	529043
Hall Monitor (qcif 176X144)	10	7.5	16	28.4	23443	28.406	23353	28.406	23207	28.416	23169	28.41	22246	28.437	21257	28.428	20849	28.411	19937	28.404	19471
			32	28.354	23258	28.352	23251	28.352	23161	28.358	22952	28.364	22299	28.354	21169	28.387	20764	28.388	20177	28.351	19712
Mother & Daughter (qcif 176X144)	24	10	16	35.26	44933	35.266	44095	35.318	42574	35.298	41767	35.279	37824	35.309	34771	35.297	32090	35.291	30500	35.295	29689
			32	35.221	44935	35.245	44220	35.13	43598	35.174	42742	35.253	37806	35.149	35311	35.25	32483	35.228	30700	35.233	29557
News (cif 352X288)	112	15	16	33.316	250972	33.314	249065	33.333	246498	33.307	244864	33.295	237268	33.298	233173	33.338	223850	33.32	218346	33.294	214368
			32	33.277	254962	33.3	251705	33.314	248899	33.3	245999	33.337	237728	33.328	231091	33.333	223800	33.321	220233	33.322	213923
News (cif 352X288)	48	7.5	16	32.514	111740	32.447	112210	32.37	112439	32.49	109331	32.504	104680	32.575	98819	32.453	100072	32.476	95912	32.463	92560
			32	32.461	112329	32.526	110114	32.472	110300	32.521	105324	32.546	102504	32.519	100358	32.436	99694	32.465	95566	32.459	93514
Silence (qcif 176X144)	24	10	16	30.969	71515	30.975	70991	30.971	71076	30.974	69530	30.989	67182	30.977	64725	30.974	61964	30.956	59383	30.985	56010
			32	30.922	72012	30.912	71912	30.93	71001	30.944	70825	30.967	67197	30.972	65169	30.964	62260	30.939	59480	30.971	56291
average				32.570	369290	32.574	367932	32.570	366656	32.577	365269	32.580	359946	32.579	355282	32.574	351103	32.571	347251	32.569	343966

most background elements move in the same direction. In the latter, the motion reflects the direction of the foreground object motion. In most talking-head video sequences, the background is relatively steady and only the foreground objects are active. Thus, they are object motion dominant video. In contrast, in most sport videos, the camera usually focuses on the athletes (objects), while the background elements move in the same direction when the camera is panning. Such videos are camera panning dominant video. Among the video sequences involved in the experiments, "flower," "coastguard," and "foreman" are camera panning dominant video sequences, while the rest are foreground object motion dominant video sequences.

From Table III, we observe that the average number of checkpoints of FAME is 30% less than MVFAST and 20% less than PMVFAST in object motion dominant video. However, the values increase to 60% less as compared MVFAST and 90% less than PMVFAST in camera panning dominant video. That implies that FAME can handle camera panning dominant video better than MVFAST and PMVFAST. This is due to the inertia temporal MV predictor that can predict the background panning motion more accurately than the spatial MV predictors.

C. *Effect of TSB Upper Bound*

We investigated the performance of the algorithm by setting deferent upper bound for TSB (see Tables IV and V). Figs. 6 and 7 show the checkpoint curves. Figs. 8 and 9 show the PSNR curves. We observe that an increase in the upper bound leads to small number of checkpoints involved. However, the PSNR also tends to decrease. The reason is that when the upper bound of the threshold is increased, more MBs are detected as stationary

TABLE VI
EFFECT OF DIFFERENT UPPER BOUND OF THS (CCIR FORMAT VIDEO)

Sequence	BR	FR	SR	THSUB = 4096		THSUB = 4608		THSUB = 5120		THSUB = 5632		THSUB = 6144		THSUB = 6656		THSUB = 7168		THSUB = 7680		THSUB = 8192	
				YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts
Cheerleader (ccir 720X480)	4000	30	64	29.320	3523243	29.307	3468725	29.306	3429603	29.296	3393062	29.291	3363067	29.289	3336576	29.285	3322757	29.280	3306738	29.281	3298434
			128	29.233	3528157	29.221	3475139	29.215	3429531	29.203	3392882	29.201	3367414	29.196	3343488	29.198	3334509	29.187	3317598	29.188	3314362
Cheerleader (ccir 720X480)	9000	30	64	33.784	3489972	33.778	3436669	33.769	3390762	33.770	3355696	33.765	3329527	33.762	3308772	33.765	3291425	33.765	3280519	33.763	3270014
			128	33.770	3492197	33.763	3441746	33.758	3395016	33.757	3360006	33.754	3331162	33.748	3309340	33.756	3290302	33.753	3281444	33.750	3271002
Flower Garden (ccir 720X480)	4000	30	64	29.230	1636181	29.215	1602768	29.192	1579797	29.164	1563982	29.139	1552338	29.116	1545368	29.098	1535569	29.083	1531822	29.068	1532217
			128	29.196	1639178	29.179	1604483	29.159	1580064	29.132	1565101	29.110	1553310	29.085	1546266	29.070	1536757	29.052	1533322	29.049	1531786
Flower Garden (ccir 720X480)	9000	30	64	33.532	1573833	33.518	1546122	33.496	1528781	33.472	1516144	33.456	1506624	33.440	1498861	33.421	1491898	33.401	1488141	33.377	1486497
			128	33.587	1573213	33.576	1546503	33.555	1528755	33.534	1514359	33.514	1505342	33.499	1499041	33.482	1491326	33.464	1487450	33.437	1484984
Average				31.457	2556997	31.445	2515269	31.431	2482789	31.416	2457654	31.404	2438598	31.392	2423464	31.384	2411818	31.373	2403379	31.364	2398662

TABLE VII
EFFECT OF DIFFERENT UPPER BOUND OF THS (QCIF AND CIF FORMAT VIDEO)

Sequence	BR	FR	SR	THSUB = 512		THSUB = 640		THSUB = 768		THSUB = 896		THSUB = 1024		THSUB = 1152		THSUB = 1280		THSUB = 1408		THSUB = 1536	
				YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts	YPSNR	ChkPts
Coastguard (cif 352X288)	112	10	16	28.13	613878	28.131	611022	28.131	607258	28.126	602476	28.123	589253	28.127	569596	28.133	547025	28.136	525919	28.124	506985
			32	28.367	597225	28.368	593862	28.365	590152	28.367	584464	28.371	567565	28.362	547292	28.362	524831	28.377	503422	28.374	483434
Coastguard (qcif 176X144)	48	10	16	30.704	109775	30.731	109060	30.707	108589	30.724	106100	30.708	99521	30.737	92484	30.709	86423	30.661	81542	30.634	76891
			32	30.696	110261	30.7	109501	30.703	108676	30.702	106502	30.709	99759	30.691	92914	30.692	87221	30.653	81921	30.653	76888
Container (qcif 176X144)	10	7.5	16	28.8	24593	28.793	24386	28.793	24285	28.793	23848	28.738	21249	28.739	21021	28.731	20453	28.731	20050	28.752	19840
			32	28.747	22432	28.747	22164	28.747	21960	28.746	21681	28.761	22201	28.761	21824	28.72	20619	28.72	20246	28.719	19835
Foreman (cif 352X288)	1024	30	16	36.23	1459818	36.223	1385029	36.207	1321331	36.194	1267151	36.189	1218164	36.168	1181002	36.157	1147596	36.151	1124664	36.137	1102359
			32	36.36	1460403	36.346	1383603	36.323	1317651	36.313	1263609	36.293	1210691	36.282	1171246	36.273	1139173	36.264	1115874	36.254	1090662
			48	36.375	1453458	36.356	1377908	36.341	1310528	36.32	1256764	36.311	1200597	36.3	1159416	36.284	1127932	36.277	1102351	36.268	1081523
Foreman (cif 352X288)	112	10	16	32.498	533517	32.512	511288	32.509	485945	32.51	460604	32.532	432897	32.516	412916	32.501	395258	32.497	384084	32.486	374810
			32	32.541	538269	32.531	516673	32.546	489536	32.553	464313	32.551	434752	32.537	415671	32.532	397456	32.523	386753	32.516	377190
Foreman (cif 352X288)	512	15	16	37.294	648365	37.281	604217	37.266	568683	37.245	539861	37.224	512836	37.209	493932	37.196	478791	37.167	468980	37.166	460792
			32	37.302	656497	37.289	610763	37.271	575251	37.254	545025	37.226	518750	37.203	498824	37.192	483760	37.18	473378	37.172	464621
			48	37.275	657297	37.27	612493	37.245	576909	37.218	547550	37.194	519585	37.179	499552	37.163	485448	37.148	474098	37.135	466134
Hall Monitor (qcif 176X144)	10	7.5	16	28.383	25215	28.391	24702	28.416	23775	28.416	23169	28.406	21936	28.379	20895	28.426	19911	28.441	19678	28.452	19179
			32	28.367	24738	28.359	24237	28.356	23722	28.358	22952	28.372	21843	28.413	20927	28.404	20380	28.424	19593	28.395	19048
Mother & Daughter (qcif 176X144)	24	10	16	35.188	44143	35.189	43592	35.182	43191	35.298	41767	35.306	38325	35.298	35272	35.294	32952	35.306	31243	35.25	29969
			32	35.117	44306	35.076	44535	35.069	43938	35.174	42742	35.29	37815	35.189	35218	35.209	33209	35.109	31716	35.176	30386
News (cif 352X288)	112	15	16	33.289	266566	33.327	255658	33.303	251583	33.307	244864	33.328	235213	33.335	225993	33.348	221052	33.328	214083	33.306	208908
			32	33.221	270078	33.329	257107	33.287	252558	33.3	245999	33.345	236123	33.307	229192	33.313	223603	33.309	216719	33.289	213150
News (cif 352X288)	48	7.5	16	32.553	116011	32.557	113284	32.49	111276	32.49	109331	32.501	102728	32.542	99062	32.539	94579	32.484	91929	32.598	87925
			32	32.491	117910	32.549	113114	32.516	111581	32.521	105324	32.534	102719	32.428	99573	32.506	94444	32.517	91954	32.511	89762
Silence (qcif 176X144)	24	10	16	30.959	74041	30.955	73259	30.958	71726	30.974	69530	30.955	66806	30.962	63873	30.964	61475	30.945	58763	30.971	55729
			32	30.955	73242	30.948	73055	30.945	72247	30.944	70825	30.942	66823	30.962	64023	30.936	62061	30.947	58831	30.96	56123
average				32.577	414252	32.582	395605	32.570	379681	32.577	365269	32.580	349090	32.568	336322	32.566	325236	32.554	316575	32.554	308839

TABLE VIII
EFFECT OF L_1 AND L_2 (CCIR FORMAT VIDEO)

Sequence			Cheerleader (ccir 720X480)		Cheerleader (ccir 720X480)		Flower Garden (ccir 720X480)		Flower Garden (ccir 720X480)		average
Bit Rate			4000		9000		4000		9000		
FR			30		30		30		30		
SR			64	128	64	128	64	128	64	128	
L₂ = 2	L₁ = 1	YPSNR	29. 329	29. 233	33. 798	33. 786	29. 25	29. 217	33. 549	33. 606	31.471
		ChkPts	3757754	3763713	3719084	3721614	1839594	1841064	1759044	1760208	2770259
L₂ = 3	L₁ = 1	YPSNR	29. 319	29. 227	33. 789	33. 769	29. 229	29. 188	33. 533	33. 587	31.455
		ChkPts	3588284	3596631	3555875	3558995	1688914	1691011	1626755	1627411	2616735
	L₁ = 2	YPSNR	29. 32	29. 231	33. 787	33. 778	29. 212	29. 176	33. 501	33. 554	31.445
		ChkPts	3611720	3617348	3575935	3576867	1841475	1841355	1776489	1776253	2702180
L₂ = 4	L₁ = 1	YPSNR	29. 307	29. 221	33. 778	33. 763	29. 215	29. 179	33. 518	33. 576	31.445
		ChkPts	3468725	3475139	3436669	3441746	1602768	1604483	1546122	1546503	2515269
	L₁ = 2	YPSNR	29. 31	29. 215	33. 778	33. 766	29. 187	29. 163	33. 482	33. 541	31.430
		ChkPts	3493382	3500597	3457556	3459970	1757880	1759205	1698860	1698028	2603185
	L₁ = 3	YPSNR	29. 312	29. 217	33. 779	33. 764	29. 177	29. 136	33. 463	33. 524	31.422
		ChkPts	3509345	3514197	3473569	3476639	1798182	1799488	1738962	1739050	2631179
L₂ = 5	L₁ = 1	YPSNR	29. 296	29. 21	33. 767	33. 751	29. 191	29. 156	33. 503	33. 56	31.429
		ChkPts	3366673	3371234	3335806	3340567	1541739	1545356	1487550	1487125	2434506
	L₁ = 2	YPSNR	29. 298	29. 21	33. 768	33. 754	29. 17	29. 134	33. 472	33. 529	31.417
		ChkPts	3391352	3396211	3357791	3359110	1696148	1697812	1641654	1640598	2522585
	L₁ = 3	YPSNR	29. 296	29. 208	33. 767	33. 754	29. 16	29. 12	33. 453	33. 516	31.409
		ChkPts	3406590	3412522	3377505	3378873	1735050	1737118	1682351	1681366	2551422
	L₁ = 4	YPSNR	29. 289	29. 196	33. 761	33. 745	29. 131	29. 089	33. 432	33. 492	31.392
		ChkPts	3422256	3429917	3390773	3392882	1755329	1756117	1701380	1701297	2568744
L₂ = 6	L₁ = 1	YPSNR	29. 283	29. 196	33. 756	33. 739	29. 162	29. 127	33. 489	33. 552	31.413
		ChkPts	3280962	3283192	3245020	3249603	1504306	1504861	1452739	1450846	2371441
	L₁ = 2	YPSNR	29. 284	29. 203	33. 754	33. 739	29. 151	29. 114	33. 452	33. 515	31.402
		ChkPts	3300388	3303941	3267009	3269673	1657198	1659037	1607173	1606191	2458826
	L₁ = 3	YPSNR	29. 286	29. 195	33. 754	33. 739	29. 125	29. 097	33. 439	33. 499	31.392
		ChkPts	3320709	3325306	3287533	3287275	1697464	1699717	1646555	1645457	2488752
	L₁ = 4	YPSNR	29. 276	29. 186	33. 748	33. 732	29. 112	29. 071	33. 419	33. 479	31.378
		ChkPts	3333727	3333510	3301818	3300361	1721546	1724187	1669040	1668692	2506610
	L₁ = 5	YPSNR	29. 264	29. 176	33. 736	33. 725	29. 064	29. 026	33. 379	33. 437	31.351
		ChkPts	3346135	3350963	3316627	3315310	1752617	1752483	1697923	1699405	2528933

TABLE IX
EFFECT OF L_1 AND L_2 (QCIF AND CIF FORMAT VIDEO)

Sequence			Coastguard (cif 352X288)		Coastguard (qcif 176X144)		Container (qcif 176X144)		Foreman (cif 352X288)			Foreman (cif 352X288)		Foreman (cif 352X288)		
Bit Rate			112		48		10		1024			112		512		
FR			10		10		7.5		30			10		15		
SR			16	32	16	32	16	32	16	32	48	16	32	16	32	48
L_2 = 2	L_1 = 1	YPSNR	28.118	28.352	30.702	30.703	28.796	28.746	36.217	36.331	36.347	32.508	32.542	37.278	37.288	37.256
		ChkPts	735275	710795	117975	117395	24863	22857	1391239	1385984	1376804	534134	539275	611992	617908	618916
L_2 = 3	L_1 = 1	YPSNR	28.12	28.36	30.723	30.71	28.793	28.746	36.201	36.32	36.335	32.512	32.543	37.25	37.254	37.239
		ChkPts	645656	622586	107573	107915	23891	21960	1303906	1295691	1287054	483229	487128	562136	567627	567619
	L_1 = 2	YPSNR	28.125	28.356	30.738	30.664	28.794	28.748	36.199	36.32	36.334	32.511	32.558	37.272	37.27	37.233
		ChkPts	672917	651268	117328	117778	24101	22130	1350755	1343067	1337398	497969	498976	586772	591910	591704
L_2 = 4	L_1 = 1	YPSNR	28.126	28.367	30.724	30.702	28.793	28.746	36.194	36.313	36.32	32.51	32.553	37.245	37.254	37.218
		ChkPts	602476	584464	106100	106502	23848	21681	1267151	1263609	1256764	460604	464313	539861	545025	547550
	L_1 = 2	YPSNR	28.126	28.361	30.734	30.667	28.794	28.746	36.199	36.313	36.332	32.501	32.538	37.254	37.256	37.227
		ChkPts	632734	612480	115752	116447	24058	22035	1318348	1310642	1304352	474808	478858	565267	569668	572221
	L_1 = 3	YPSNR	28.115	28.363	30.689	30.674	28.794	28.748	36.167	36.271	36.296	32.465	32.511	37.192	37.208	37.176
		ChkPts	649868	628342	118820	118982	24051	21954	1346311	1340443	1334984	487647	489526	584255	587841	589302
L_2 = 5	L_1 = 1	YPSNR	28.123	28.356	30.724	30.702	28.793	28.746	36.179	36.311	36.318	32.519	32.548	37.244	37.246	37.207
		ChkPts	585825	567565	105563	105940	23753	21639	1248270	1242046	1235451	448251	451819	527209	533340	534830
	L_1 = 2	YPSNR	28.125	28.36	30.734	30.667	28.794	28.746	36.197	36.3	36.318	32.489	32.531	37.25	37.253	37.216
		ChkPts	612278	595166	115313	115983	23961	21895	1296155	1292141	1286081	465389	466332	553026	557356	560264
	L_1 = 3	YPSNR	28.113	28.354	30.689	30.674	28.794	28.748	36.166	36.278	36.282	32.472	32.515	37.202	37.206	37.17
		ChkPts	630413	611739	118319	118473	23954	21809	1327481	1321576	1315668	475955	478803	572482	576229	577974
	L_1 = 4	YPSNR	28.101	28.34	30.694	30.688	28.794	28.748	36.145	36.263	36.272	32.468	32.508	37.147	37.165	37.133
		ChkPts	641119	622591	118605	118585	23968	21799	1338918	1330713	1325058	481699	483765	581193	584080	585206
L_2 = 6	L_1 = 1	YPSNR	28.124	28.359	30.724	30.702	28.793	28.746	36.186	36.305	36.312	32.499	32.541	37.233	37.237	37.207
		ChkPts	577833	560856	105493	105862	23745	21631	1236193	1232482	1224979	442565	445746	520641	526242	527998
	L_1 = 2	YPSNR	28.13	28.364	30.734	30.667	28.794	28.746	36.199	36.307	36.316	32.493	32.527	37.238	37.232	37.219
		ChkPts	604421	588327	115195	115888	23953	21863	1284418	1278620	1273811	458169	461481	546578	550800	552835
	L_1 = 3	YPSNR	28.114	28.35	30.689	30.674	28.794	28.748	36.156	36.272	36.28	32.467	32.513	37.183	37.183	37.164
		ChkPts	623398	605075	118268	118383	23941	21777	1317229	1309809	1305548	470141	472432	566774	569489	571285
	L1 = 4	YPSNR	28.105	28.34	30.694	30.688	28.794	28.748	36.14	36.251	36.263	32.458	32.493	37.14	37.152	37.123
		ChkPts	632758	615701	118542	118512	23955	21767	1326713	1319572	1313083	474676	477697	573376	578095	577915
	L1 = 5	YPSNR	28.104	28.332	30.681	30.683	28.792	28.748	36.116	36.234	36.242	32.459	32.497	37.111	37.123	37.097
		ChkPts	639535	622459	118764	118769	24083	21789	1337421	1331099	1324105	479128	481318	580036	584063	585249

TABLE IX (*Continued*)
EFFECT OF L_1 AND L_2 (QCIF AND CIF FORMAT VIDEO)

Sequence			Hall Monitor (qcif 176X144)		Mother & Daughter (qcif 176X144)		News (cif 352X288)		News (cif 352X288)		Silence (qcif 176X144)		average
Bit Rate			10		24		112		48		24		
FR			7.5		10		15		7.5		10		
SR			16	32	16	32	16	32	16	32	16	32	
L_2 = 2	L_1 = 1	YPSNR	28.449	28.364	35.28	35.175	33.294	33.302	32.45	32.489	30.962	30.953	32.579
		ChkPts	26840	26446	45504	45736	277987	278975	127028	125706	75913	76528	413003
L_2 = 3	L_1 = 1	YPSNR	28.41	28.356	35.227	35.151	33.282	33.34	32.508	32.539	30.978	30.936	32.576
		ChkPts	24328	24148	42886	43574	256166	255425	114637	112154	71827	72427	379231
	L_1 = 2	YPSNR	28.415	28.356	35.216	35.189	33.327	33.34	32.573	32.558	30.953	30.959	32.584
		ChkPts	24977	24732	44879	44243	261388	263085	115244	114811	73842	73646	393538
L_2 = 4	L_1 = 1	YPSNR	28.416	28.358	35.298	35.174	33.307	33.3	32.49	32.521	30.974	30.944	32.577
		ChkPts	23169	22952	41767	42742	244864	245999	109331	105324	69530	70825	365269
	L_1 = 2	YPSNR	28.416	28.357	35.276	35.12	33.345	33.308	32.478	32.527	30.969	30.976	32.576
		ChkPts	23724	23521	42918	44083	249800	253784	111681	110595	70936	71823	380022
	L_1 = 3	YPSNR	28.368	28.321	35.234	35.264	33.271	33.216	32.442	32.506	30.981	30.932	32.550
		ChkPts	24193	24175	43592	42240	257765	262545	114582	113184	70964	71709	389470
L_2 = 5	L_1 = 1	YPSNR	28.414	28.358	35.225	35.116	33.321	33.245	32.437	32.405	30.95	30.948	32.560
		ChkPts	22674	22661	41318	42022	238045	244752	108572	109201	68556	68836	358256
	L_1 = 2	YPSNR	28.414	28.355	35.219	35.166	33.3	33.275	32.563	32.484	30.971	30.964	32.570
		ChkPts	23311	23203	42616	42802	247819	250545	108857	110855	69869	69415	372943
	L_1 = 3	YPSNR	28.39	28.304	35.184	35.224	33.259	33.285	32.538	32.522	30.994	30.97	32.556
		ChkPts	23551	23829	43659	42282	254002	251566	110387	111186	69424	69793	382106
	L_1 = 4	YPSNR	28.369	28.279	35.221	35.273	33.253	33.269	32.524	32.498	30.942	30.946	32.543
		ChkPts	23492	24214	42832	41755	258126	254503	113033	113362	70615	70503	386239
L_2 = 6	L_1 = 1	YPSNR	28.414	28.371	35.194	35.163	33.307	33.277	32.546	32.503	30.981	30.957	32.570
		ChkPts	22547	22347	41065	41257	237123	238574	104727	105096	67232	67581	354159
	L_1 = 2	YPSNR	28.414	28.371	35.26	35.129	33.289	33.297	32.513	32.555	30.976	30.976	32.573
		ChkPts	23072	22818	42191	42894	243205	244001	109215	107523	68359	68342	368666
	L_1 = 3	YPSNR	28.413	28.348	35.178	35.229	33.273	33.308	32.53	32.548	30.973	30.958	32.556
		ChkPts	23315	23346	42744	41948	249135	247732	109483	108768	68588	69166	378241
	L1 = 4	YPSNR	28.374	28.343	35.199	35.259	33.282	33.292	32.431	32.554	30.957	30.962	32.543
		ChkPts	23117	22970	43092	42105	251956	251799	114036	109935	69693	69760	382118
	L1 = 5	YPSNR	28.39	28.36	35.273	35.248	33.292	33.197	32.507	32.53	30.973	30.95	32.539
		ChkPts	23407	22846	42478	41916	254376	259427	112078	111538	69910	70289	385670

TABLE X
QUALITATIVE COMPARISON OF MVFAST, PMVFAST, AND FAME ALGORITHM

	MVF	PMV	FAME
Threshold comparisons	No threshold or one optional threshold	A set of compulsory thresholds. Coding performance and sensitivity of PMVFAST using these thresholds for the video sequences and encoding conditions outside the MPEG-4 test set has *not* been studied and verified.	A set of adaptive thresholds.
Memory	No need to store either search points or motion vectors	Memory is compulsory. Up to 4 Mbytes of memory for a search window size of ±1024 to store search points and up to 1.3 KB for storing MVs	Memory is compulsory. Up to 4 Mbytes of memory for a search window size of ±1024 to store search points and extra memory for storing MVs of reference frame.
Objective quality (PSNR)	On average, about 0.2 dB less than Full Search. (0.033dB better than full search in our experiment)	On average, about 0.1dB less than Full Search (0.9dB less than full search in our experiment)	Similar to full search in our experiment
Speed-up		Claimed to be about 50% faster than MVFAST (Not better than MVFAST in our experiments)	About 50% faster than MVFAST and PMVFAST
Motion inertia	Not supported	Not supported	supported

MBs. Consequently, the number of checkpoints decreases. This is a tradeoff situation as some MBs can be wrongly taken as stationary, which lowers the video quality. The selection of the upper bound of the threshold would depend on the user requirements in terms of the speed and quality. Here UB is the upper bound of TSB.

D. Effect of THS Upper Bound

Different upper bounds for THS were used to investigate its effects (see Table VI and VII). Figs. 10 and 11 show the checkpoint curves. Figs. 12 and 13 show the PSNR curve. When we increase the upper bound of THS, both the numbers of checkpoints and the PSNR value decrease. This observation is similar to the effect of the upper bound of TSB. When the upper bound of THS is increased, the algorithm terminates search in more MBs with larger SAD values. The match is not considerably accurate but is much faster. Hence, the numbers of checkpoints as well as the PSNR values decrease. Here UB is the upper bound of THS.

E. The Effect of L_1 and L_2

We used different L_1 and L_2 values to investigate their effect on the performance. From Tables VIII and IX, we observed that by fixing the value of L_2, generally, both the speed and PSNR performance tend to decrease when the value of L_1 increases. Again, by fixing L_1 to 1, we find that generally when the value of L_2 increases, the speed increases but the PSNR drops. Thus, there is a tradeoff between speed and visual quality. We choose $L_1 = 1$ and $L_2 = 4$ in the experiments.

F. Qualitative Comparison

We summarize the qualities of MVFAST, PMVFAST, and the proposed FAME algorithm. The qualitative comparison is made in Table X.

G. Conclusions

FAME outperforms MVFAST with much faster speed while yielding the similar picture quality with full search. MVFAST causes steep degradation when coding fast motion sequences, like Stefan, in our experiments. However, FAME not only achieves the similar quality with full search, but also uses less checking points compared to MVFAST. The experiments show that FAME can cope well with both large dynamic motion variation sequence and simple uniform motion video. FAME is very suitable for real-time high quality MPEG-4 video encoding.

REFERENCES

[1] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bitrate reduction of MPEG-2 bit streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953–967, Dec. 1998.

[2] M. Bierling, "Displacement estimation by hierarchical blockmatching," *SPIE Vis. Commun. Image Process.*, pp. 942–951, May 1998.

[3] Y. L. Chan and W. C. Siu, "Adaptive multiple-candidate hierarchical search for block matching algorithm," *IEE Electron. Lett.*, vol. 31, no. 19, pp. 1637–1639, Sep. 1995.

[4] L. G. Chen, W. T. Chen, Y. S. Jehng, and T. D. Chuieh, "An efficient parallel motion estimation algorithm for digital image processing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 4, pp. 378–384, Dec. 1991.

[5] C. K. Cheung and L. M. Po, "A hierarchical block motion estimation algorithm using partial distortion measure," in *Proc. ICIP'97*, vol. 3, 1997, pp. 606–609.

[6] J. Feng, K. T. Lo, H. Mehrpour, and A. E. Karbowiak, "Adaptive block-matching motion estimation algorithm for video coding," *IEE Electron. Lett.*, vol. 31, no. 18, pp. 1542–1543, 1995.

[7] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, no. 7, pp. 950–953, Jul. 1990.

[8] P. I. Hosur and K. K. Ma, "Motion vector field adaptive fast motion estimation," presented at the *Second Int. Conf. Inf., Commun., Signal Process.*, Singapore, Dec. 1999.

[9] ——, "Report on performance of fast motion estimation using motion vector field adaptive search technique (MVFAST)," ISO/IEC JTC1/SC29/WG11 M5453, Dec. 1999.

[10] ——, "Performance report of motion vector field adaptive search technique (MVFAST)," ISO/IEC JTC1/SC29/WG11 M5851, Noordwijkerhout, Netherlands, Mar. 2000.

[11] Z. He and M. L. Liou, "A high performance fast search algorithm for block matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 5, pp. 826–828, Oct. 1997.

[12] ——, "Design of fast motion estimation algorithm based on hardware consideration," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 5, pp. 819–823, Oct. 1997.

[13] *Optimization Model Version 1.0*, ISO/IEC JTC1/SC29/WG11 N3324, Mar. 2000.

[14] *Information Technology—Generic Coding of Audio Visual Object*, Oct. 1998.

[15] *Experimental Conditions for Evaluating Encoder Motion Estimation Algorithms*, Dec. 1999.

[16] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, no. 12, pp. 1799–1808, Dec. 1981.

[17] H. M. Jung, D. D. Hwang, C. S. Park, and H. S. Kim, "An annular search algorithm for efficient motion estimation," in *Proc. Int. Picture Coding Symp.*, 1996, pp. 171–174.

[18] T. Koga, K. Ilinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov. 1981, pp. G5.3.1–G5.3.5..

[19] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation*. Norwell, MA: Kluwer, 1999.

[20] L. W. Lee, J. F. Wang, J. Y. Lee, and J. D. Shie, "Dynamic search-window adjustment and interlaced search block-matching algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 1, pp. 85–87, Feb. 1993.

[21] X. Lee and Y. Q. Zhang, "A fast hierarchical motion-compensation scheme for video coding using block-feature matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 6, pp. 627–635, Dec. 1996.

[22] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for fast motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438–442, Aug. 1994.

[23] B. Liu and A. Zaccartin, "New fast algorithms for estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 2, pp. 148–157, Apr. 1993.

[24] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block-based motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419–422, Aug. 1996.

[25] A. Netravali and B. Haskell, *Digital Pictures Representation and Compression*. New York: Plenum, 1988.

[26] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast blockmatching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, Jun. 1996.

[27] A. Puri, H. M. Hang, and D. L. Schilling, "An efficient block matching algorithm for motion compensated coding," in *Proc. IEEE ICASSP*, 1987, pp. 2.4.1–25.4.4.

[28] Y. Q. Shi and X. Xia, "A thresholding multiresolution block matching algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 437–440, Apr. 1997.

[29] X. Song, T. Chiang, and Y. Q. Zhang, "A scalable hierarchical motion estimation algorithm for MPEG-2," in *Proc. ICIP*, 1998, pp. IV126–IV129.

[30] B. C. Song and J. B. Ra, "A hierarchical block matching algorithm using partial distortion criteria," in *Proc. VCIP Vis. Commun. Image Process.*, San Jose, CA, 1998, pp. 88–95.

[31] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. Com-33, no. 8, pp. 888–896, Aug. 1985.

[32] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–377, Aug. 1998.

[33] A. M. Tourapis, O. C. Au, and M. L. Liou, "Fast block-matching motion estimation using advanced predictive diamond zonal search (APDZS)," ISO/IEC JTC1/SC29/WG11 MPEG2000/M5865, Noordwijkerhout, The Netherlands, Mar. 2000.

[34] ——, "Fast block-matching motion estimation using predictive motion vector field adaptive search technique (PMVFAST)," ISO/IEC JTC1/SC29/WG11 MPEG2000/M5866, Noordwijkerhout, The Netherlands, Mar. 2000.

[35] A. M. Tourapis, O. C. Au, M. L. Liou, G. Shen, and I. Ahmad, "Optimizing the MPEG-4 encoder—advanced diamond zonal search," in *Proc. Int. Symp. Circuits Syst. (ISCAS)*, Geneva, Switzerland, Jun. 2000, pp. 674–680.

[36] J. B. Xu, L. M. Po, and C. K. Cheung, "A new prediction model search algorithm for fast block motion estimation," in *IEEE Int. Conf. Image Process.*, 1997, pp. 610–613.

[37] B. Zeng, R. Li, and M. L. Liou, "Optimization of fast block motion estimation algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 6, pp. 833–844, Dec. 1997.

[38] W. Zheng, I. Ahmad, and M. L. Liou, "Adaptive motion search with elastic diamonds for MPEG-4 video encoding," in *Proc. Int. Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 377–380.

[39] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

Ishfaq Ahmad (S'88–M'92–SM'03) received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 1985, the M.S. degree in computer engineering, and the Ph.D. degree in computer science, both from Syracuse University, Syracuse, NY, in 1987 and 1992, respectively.

He is currently a Full Professor of Computer Science and engineering in the Computer Science and Engineering Department, University of Texas (UT) at Arlington. Prior to joining UT Arlington, he was an associate professor in the Computer Science Department at Hong Kong University of Science and Technology (HKUST), Hong Kong. At HKUST, he was also the Director of the Multimedia Technology Research Center, an officially recognized researh center that he conceived and built from scratch. The center was funded by various agencies of the Government of the Hong Kong Special Administrative Region as well as local and international industries. With more than 40 personnel including faculty members, postdoctoral fellows, full-time staff, and graduate students, the center engaged in numerous research and development projects with academia and industry from Hong Kong, China, and the U.S. Particular areas of focus in the center are video (and related audio) compression technologies, videotelephone and conferencing systeme. The center commercialized several of its technologies to its industrial partners world wide. His recent research focus has been on developing parallel programming tools, scheduling and mapping algorithms for scalable architectures, heterogeneous computing systems, distributed multimedia systems, video compression techniques, and web management. His research work in these areas is published in over 125 technical papers in refereed journals and conferences.

Dr. Ahmad has recevied Best Paper Awards at Supercomputing'90 (New York), Supercomputing'91 (Albuquerque), and the 2001 International Conferene Parallel Processing (Spain). He has participated in the organization of several international conferences and is an Associate Editor of *Cluster Computing, Journal of Parallel and Distributed Computing,* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *IEEE Concurrency*, and *IEEE Distributed Systems Online*.

Weiguo Zheng (M'00) received the B.E., M.S., and Ph.D. degrees in electrical engineering from Zhongshan (Sun Yat-sen) University, Guangzhou, China, in 1992, 1995, and 1998 respectively.

He is currently an Audio Firmware Manager of Sigma Designs, Inc., Milpitus, CA, Prior to joining Sigma Designs, he was an Assistant Professor in the Electrical Engineering Department, Zhongshan University, and an Associate Researcher in the Multimedia Technology Research Center and Computer Science Department of the Hong Kong University of Science and Technology, Hong Kong, SAR. His research interest is multimedia signal processing.

Jiancong Luo (S'04-M'05) received the B.S. and M.S. degrees in electrical engineering from Zhongshan (Sun Yat-sen) University, Guangzhou, China, in 1997 and 2000, respectively. From 2000 to 2001, he was a Ph.D. student in computer science at Hong Kong University of Science and Technology, Hong Kong, SAR. He received the Ph.D. degree in computer science and engineering from the University of Texas at Arlington in August 2005.

He is currently a Member of Technical Staff with Thomson Corporate Research, Princeton, NJ. His research interests include video compression, wireless multimedia communication, and 2-D/3-D image and video processing.

Ming Liou (S'61–M'63–SM'78–F'79–LF'00) received the B.S. degree from National Taiwan University, the M.S. degree from Drexel University, Philadelphia, PA, and the Ph.D. degree from Stanford University, Stanford, CA, in 1956, 1961, and 1964, respectively, all in electrical engineering. Beginning in 1963, he was with AT&T Bell Laboratories as a Member of Technical Staff, where he held various supervisory positions and researched numerical analysis, system theory, FM distortion analysis, and computer-aided design of communication circuits and systems, including circuits containing periodically operated switches. From 1984 to 1992, he was a Director at Bellcore, Red Bank, NJ, conducting research in data transmission, digital subscriber line transceivers, and video technology. He joined the faculty of the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong Special Administrative Region of China, as a Professor in October 1992, and was appointed as Director of Hong Kong Telecom Institute of Information Technology in January 1993. His current research interests include very low bit-rate video, motion estimation techniques, video over wireless, HDTV, VLSI architecture, implementation of signal processing systems for visual applications and information technology. Dr. Liou has served in various capacities, including Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS (CAS) from 1979 to 1981, President of the IEEE CAS Society, the founding Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 1991 to 1995, Co-Guest Editor of the Special Issue on Advances in Image and Video Compression of the PROCEEDINGS OF THE IEEE in February 1995, and Co-General Chair for the IEEE International Symposium on Circuits and Systems, Hong Kong, in June 1997. He has published numerous papers in various fields and was the recipient of the IEEE CAS Society Special Prize Paper Award in 1973 and the Darlington Prize Paper Award in 1977. In recognition of his outstanding professional and technical contributions, he was the recipient of the IEEE CAS Distinguished Service Award in 1991, the IEEE CAS Golden Jubilee Medal in 1999, and the IEEE Millennium Medal and IEEE Circuits and Systems Society Mac Van Valkenburg Award in 2000. He is a member of Sigma Xi, Eta Kappa Nu, Phi Tau Phi, and a Fellow the Hong Kong Institution of Engineers.