Adaptive Double-Layered Initial Search Pattern for Fast Motion Estimation

Chong-Shou Yu and Shen-Chuan Tai, *Senior Member, IEEE*

***Abstract*—Multimedia communication relies on data compression technology to reduce the data bits of transmission. Motion compensation is the key function in exploiting temporal redundancy for compression in most video coding standards. For example, fixed search pattern motion estimation algorithms such as hexagonal search (HEXBS) usually spend extra search steps to confirm near-zero motion vectors. Divide and conquer methods such as the efficient three-step search (E3SS) are not optimized for the probability model of motion vectors. This paper proposes a pair of complementary double-layered (inner layer and outer layer) initial search patterns to reduce computational complexity. The inner-layer search is applied first and tests for small motion. Afterwards, the outer layer search serves as a guard line to catch large motion. It is used only when the inner search layer fails to find a good solution. Experimental results of motion estimation on various QCIF/CIF video sequences show that the proposed algorithm achieves image quality similar to diamond search but with the search point cost as low as cross-diamond-hexagonal search.**

***Index Terms*—Divide and conquer strategy, motion estimation, video coding.**

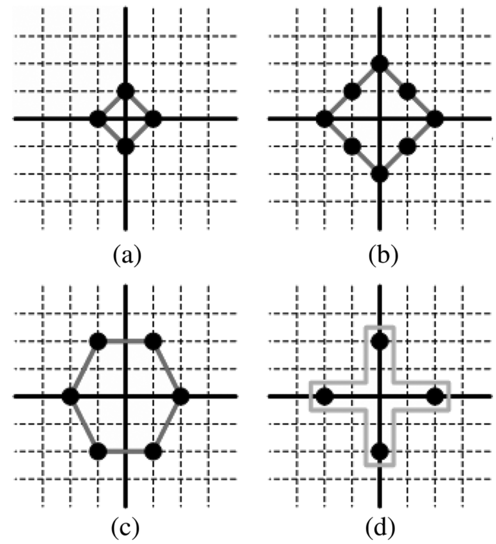


Fig. 1. Search patterns: (a) small diamond search pattern (SDSP); (b) large diamond search pattern (LDSP); (c) hexagonal search pattern; (d) large cross search pattern.

I. INTRODUCTION

MOTION estimation is the most time-consuming module in current video coding standards such as MPEG-1, 2, and 4 [1]–[3]. During the encoding of video sequences every picture is divided into macroblocks. The motion of these macroblocks is estimated by using the previous frame as a reference image. The movements of these macroblocks are then represented as a vector, which indicates the displacement between the current frame and the reference frame. Traditionally, all possible spatial displacement shall be checked to find out the correct motion vector. Although this process benefits the compression by exploiting temporal correlation between two video frames, searching all possible displacements takes extreme high computation.

Many fast search algorithms [4]–[12] have been proposed to speedup the motion estimation process. Diamond search (DS) [4] and hexagonal search (HEXBS) [5] are popular since they are easy for hardware implementation and perform reasonably in most cases. Diamond search combines two search patterns, the small diamond search pattern (SDSP) and the large diamond search pattern (LDSP) shown in Fig. 1(a) and (b). HEXBS is a hexagonal search pattern that works better than DS for videos with more oblique motion. This pattern is shown in Fig. 1(c). Enhanced HEXBS (E-HEXBS) [6] further reduces the searching cost with additional regulation.

If the motion is quasi-stationary, DS and E-HEXBS catch the correct vector within a few search points. However, if the vector with minimum distortion is located far from the center of a search window, additional search points are required to complete the search. E-HEXBS is faster than DS on some video sequences with highly inconsistent motion. However DS and E-HEXBS start their search from the search center and move only a little for each step so that they cannot catch large motion within a few steps.

Dual cross search (DCS) [7] is another fast search pattern recently published. It uses two cross search patterns, which have different sizes. One is a 5×5 cross search pattern as shown in Fig. 1(d), and the other is a 3×3 pattern that is composed of five search points identical to SDSP. Because of taking different sizes of patterns for center clustered motion vectors, DCS is very fast in most cases. However, while the motion goes to an oblique direction, DCS usually only finds inferior solutions and therefore is not suggested for videos with large motion. Adaptive dual cross search (ADCS) [7] applies an early termination rule and reduces necessary search points significantly, but the weakness of DCS still exists.

Manuscript received February 28, 2005; revised February 2, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lap-Pui Chau.

The authors are with the Data Compression and Multimedia Communication Laboratory, Department of Electrical Engineering, National Cheng-Kung University, Tainan 701, Taiwan, R.O.C. (e-mail: ytz@lily.ee.ncku.edu.tw; sctai@mail.ncku.edu.tw).

Digital Object Identifier 10.1109/TMM.2006.884635

TABLE I
VIDEO SEQUENCES USED FOR STATISTICS AND EXPERIMENTS

Frame Format	Squence	No. of Frames	Frame Format	Squence	No. of Frames
CIF (352x288)	Akiyo	300	CIF (352x288)	Singer	250
	Basket	498		Stefan	300
	Bus	150		Table Tennis	300
	Caona	220		Tempete	260
	Coastguard	300		Vectra	300
	Dancer	250	QCIF (176x144)	Carphone	382
	Football	260		Coastguard	300
	Foreman	300		Foreman	300
	Hall Monitor	300		Hall Monitor	330
	Kiel	250		Mobile	300
	Mobile	300		Stefan	300
	Silent Voice	300		Table Tennis	300

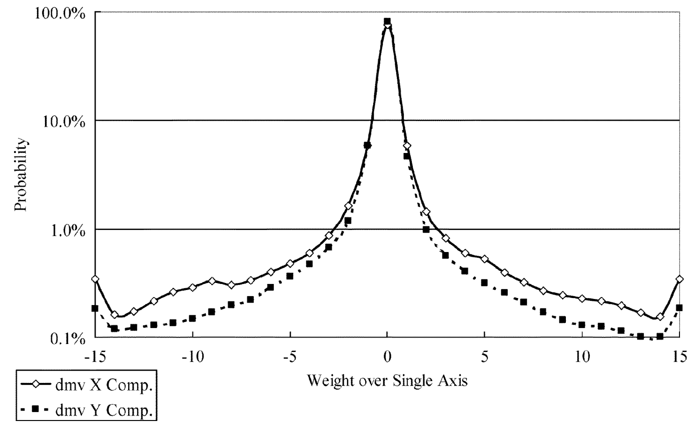


Fig. 3. Histogram of horizontal and vertical components of differential motion vectors.

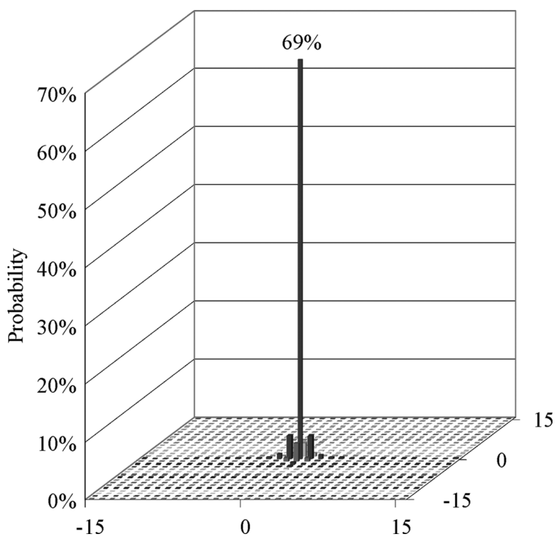


Fig. 2. Histogram of differential motion vectors.

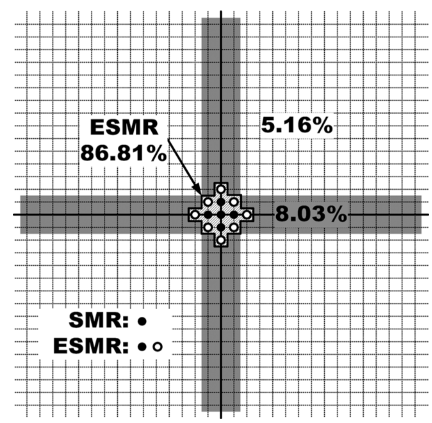


Fig. 4. Distribution map of differential motion vectors.

On the basis of DS, the cross-diamond-hexagonal search (CDHS) [8] uses hierarchy pattern structure to improve the search efficiency. With cross initial search pattern and conditional change of pattern shape, CDHS is fast and accuracy in most cases. The CDHS results in degraded performance only when the video sequence has large motions.

The efficient three-step search (E3SS) [9] is an alternate implementation of the two-dimensional logarithmic algorithm (2DLOG) [10]. It uses SDSP (small diamond search pattern [4]) at the center of the search window and adopts a divide-and-conquer strategy by searching eight spatial median points. Overall E3SS can achieve good visual quality because of its wide-spreading search points. One-dimensional logarithmic search, which is also known as binary search, divide the search range into two parts on each step. Usually the chosen search point locates on the middle of whole search range which makes the divided two parts are in equal size. However, search time can only be optimized when the search result has the same probability to fall in each part. E3SS implements the 2-D logarithmic search by taking the midpoints between the center and search window boundary instead of taking by statistical probability. Therefore E3SS is suboptimal only.

Searching the statistical median points with a divide and conquer strategy could possibly be the fastest method available, but two important issues arise. The first issue is about maintaining the statistic model so that it can be adopted for different video sequences. The second issue is that deriving the statistical median is computationally intensive, which leads to the question of whether there is any method to reduce the time cost. These issues will be addressed below with a set of pseudo statistical medians being adjusted during the process of motion estimation.

This paper will propose a novel pattern for the initial search step. We try to estimate small motions with a minimum of search points since they occur much more frequently than large motions. When it is determined that the motion isn't small during the early search, the pseudo statistical median search points are then used to locate the large motion. The rest of this paper is organized as follows. Section II reviews the statistics of motion vectors as estimated by exhaustive search. With the statistic numbers, various search pattern suited for different motion characteristics are introduced. The proposed adaptive double-layered initial search pattern (ADLISP) is introduced

TABLE II
VIDEO SEQUENCES STATISTICS

	Squence	P_{ESMR}	AM_X	AM_Y
CIF	Akiyo	99.95%	1.22	2.19
	Basket	92.27%	2.71	3.27
	Bus	78.17%	6.25	1.70
	Caona	70.04%	6.48	2.75
	Coastguard	98.30%	3.97	2.00
	Dancer	62.77%	5.42	4.59
	Football	57.32%	5.95	5.11
	Foreman	84.85%	4.63	4.18
	Hall Monitor	89.07%	2.68	5.18
	Kiel	96.44%	5.45	3.89
	Mobile	96.89%	6.92	2.14
	Silent Voice	92.91%	4.17	4.20
	Singer	99.25%	1.83	2.89
	Stefan	71.30%	7.05	2.47
	Table Tennis	87.69%	4.62	3.93
	Tempete	90.65%	6.28	4.11
	Vectra	75.84%	7.72	2.71
QCIF	Carphone	86.14%	4.22	2.97
	Coastguard	99.55%	4.23	5.13
	Foreman	88.60%	3.99	3.39
	Hall Monitor	99.28%	0.80	3.16
	Mobile	99.72%	7.78	0.45
	Stefan	82.73%	6.33	1.13
	Table Tennis	90.18%	3.94	3.56

P_{ESMR}: Probability that motion vector of full search locates on ESMR

AM_X: Average movement of horizontal component of vectors that not on ESMR

AM_Y: Average movement of vertical component of vectors that not on ESMR

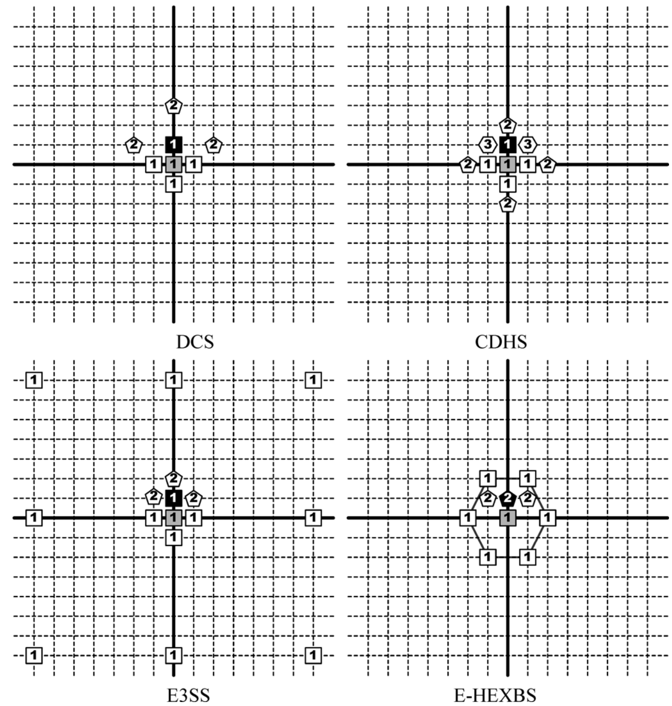


Fig. 5. Search points used for DCS, CDHS, E3SS and E-HEXBS while motion vector locates on (0, 1). The search range is 15.

in Section III. Section IV presents simulation results and comparison with other search patterns. The final conclusion will be presented in Section V.

II. SEARCH PATTERN EFFICIENCY AND STATISTICS OF MOTION VECTORS

The statistical features of motion vectors are important for designing search pattern. In order to analyze the characteristics of motion vectors, we performed exhaustive motion search on several different video sequences with MPEG-4 motion prediction algorithm.[1]

Table I lists the test sequences used in the analysis. Some of them have large and inconsistence motion, such as Foreman, Basket Ball, and Stefan. Some have small and consistence motion, such as Akiyo and Coastguard. Experiments on these sequences can cover most typical motion characteristics in normal video.

The resulting differential motion vector histogram is shown in Fig. 2. The histograms of the horizontal and vertical components of the differential motion vectors are shown in Fig. 3. Note that the histogram axis in Fig. 3 is logarithmic.

A. Small Motion

We first simply define the five center points: (0, 0), ($\pm$1, 0), (0, $\pm$1), as SMR (small motion region). With 8 additional neighboring search points, ($\pm$2, 0), ($\pm$1, $\pm$1) and (0, $\pm$2), the total of 13 search points is defined as ESMR (extended small motion region). The search points of SMR and ESMR are shown on Fig. 4.

According to the statistics, the $\boldsymbol{dMVs}$ (differences between estimated motion vectors and predicted vectors) are clustered tightly at the center of the search window, i.e., more than 86% of the $\boldsymbol{dMVs}$ are estimated to be located in ESMR. Especially the zero motion cases dominate 69% of all $\boldsymbol{dMVs}$. For detailed observation the probability of searches located in ESMR for each test sequences are listed in Table II as $\mathrm{P_{ESMR}}$. This extremely significant clustering characteristic is the major consideration of pattern specified fast motion search algorithms.

Due to this characteristic, a good search pattern shall use minimum number of search points to determine whether the motion is in the small motion area, especially (0, 0), which is the most common cases. For possible large motion cases, many algorithms use pattern that roughly covers 5 $\times$ 5 center search area. These patterns are designed to catch large motions faster while not over-expanding their size. However, the enlarged pattern size still causes additional computation load on the nonzero small motion cases. To determine whether the motion vector is located on (0, 1), DCS must check 8$\sim$11 search points, CDHS must check 11 search points, E3SS must check 16 points and E-HEXBS must check ten points. The necessary search points of these algorithms are shown on Fig. 5.

Because searching on a pattern's first-step search points is unavoidable, the minimum necessary computational cost of a search pattern is directly related to the number of their first-step search points. SDSP has only five search points, and therefore it has lower unavoidable computational cost than DCS, CDHS,

[1] The predicted motion vector of current encoding macroblock is the median of the vectors of three neighboring macroblocks: upper macroblock, upper-right macroblock and left macroblock.

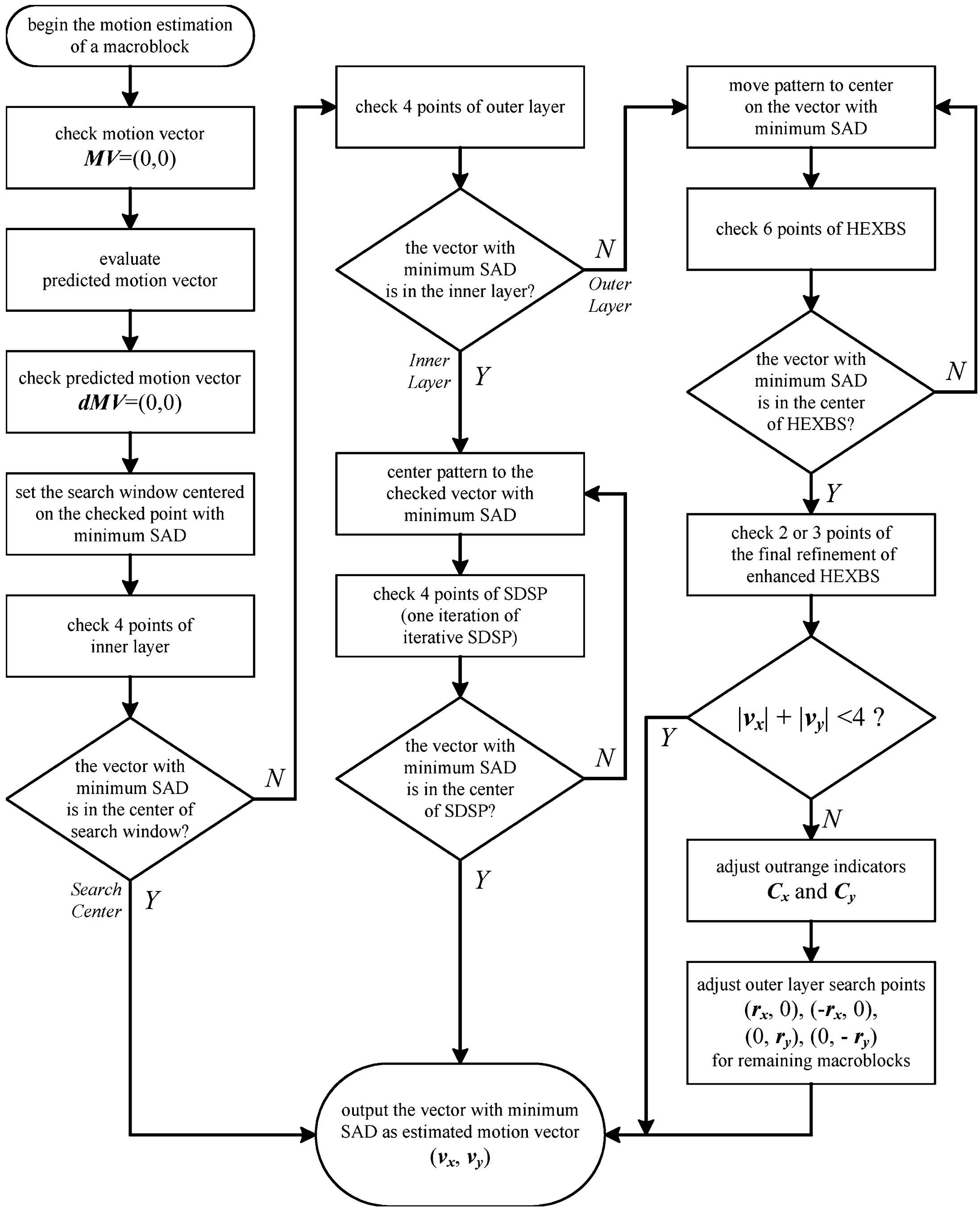


Fig. 6. System diagram of ADLISP.

E3SS, and E-HEXBS. If only iterative SDSP[2] is used for the motion search, with disregarding the low-probability large motion cases, the vector can be estimated in SMR after checking only eight points. This is how E3SS improves search efficiency impressively over 3SS [14], which only use the spatial median search points. With the concept of separating search points into small motion area and large motion area in E3SS, the SDSP proves its efficiency for small motion search. On the same concept, the proposed pattern combines SDSP for small motion together with revised version of E3SS's large motion search points.

[2]Check four search points of SDSP; move SDSP to center on newly found best matching position. Repeat this until best position no longer changes.

B. Large Motion

Using iterative SDSP at the center of the search window is fast for small motion but not for low-probability large motion cases. E3SS catches large motion by searching eight spatial median points at the first step. It is costly to calculate those additional points, especially they are rarely happens in small motion cases.

There are two ways to reduce the additional computation loads. The first is to check these additional spatial median

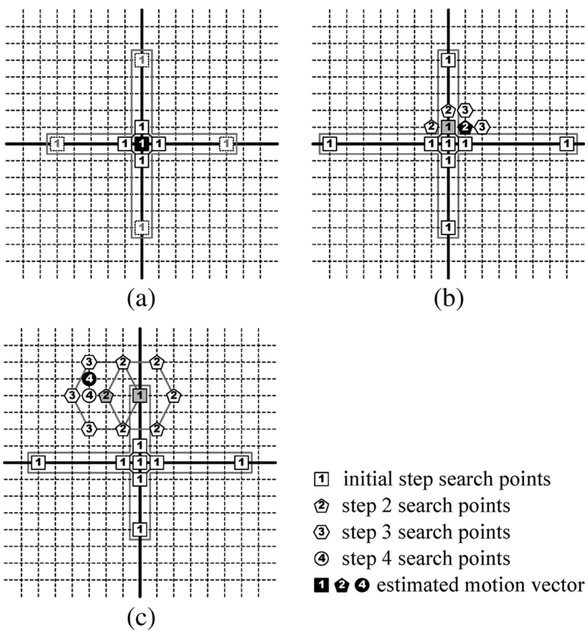


Fig. 7. Adaptive double-layered initial search pattern. (a) First-step-stop example with (0, 0) motion vector estimated $r_x = 5$ and $r_y = 5$ in this example; (b) SDSP process when initial search result falls on inner layer, $r_x = 7$ and $r_y = 5$ in this example; (c) 6-side fast inner search process when initial search result falls on outer layer, $r_x = 6$ and $r_y = 4$ in this example.

points only when necessary: Perform E3SS to check center SDSP first, and check the additional eight search points later only when motion is confirmed not (0, 0). Since half of the true motion vectors are static (refer to Fig. 2), this change could save much computation power. The second method is using reduced number of additional spatial median search points.

To determine the optimal number of search points for large motions, a further study on the statistical model is necessary. The statistics of full search shows that if the motion vectors located on SMR are omitted, the majority of remaining motion vectors is mostly found on purely horizontal or vertical directions. In other words, oblique motion vectors are rare. This is shown in Fig. 4. Therefore, oblique search points far from the search center can be omitted during the first search step with only small quality losses. As long as the later steps can take the responsibility to check these rarely used oblique search points, it will speedup the whole search. In conclusion, it is better to keep the four search points on horizontal and vertical axes only.

The positions of the four remaining search points are also need to be revised. According to the histogram in Fig. 3, variation of the horizontal component of the motion vector is larger than the vertical component. In other words, if the video frame is not stationary, most real video sequences will have motion along the horizontal axis. This means that if a 2-D logarithmic search is used, to optimize the search the statistical horizontal and vertical median points will not be located at the same distance from the center of search window. Average movement[3] of full search motion vectors can be used to derive best statistical median points. Note that if iterative SDSP is used to catch small motion in search center, the search points in ESMR will mostly be excluded from the 2-D logarithmic search range. The average movement of the remaining motion vectors is 5.53 in horizontal and 3.68 in vertical. Detail average movement values of each sequence are listed in Table II.

The statistical median points that decided by average movement could serve as a good checkpoint for a divide-and-conquer strategy. The discussion above gives good search points for an initial search step to catch large motion: ($\pm AM_X$, 0) and (0, $\pm AM_Y$), where AM_X and AM_Y are average movements of horizontal and vertical large motion vectors respectively. Our previous work DLISP [15] adopted a fixed asymmetric large cross pattern (6, 0), (−6, 0), (0, 4), (0, −4) and achieved moderate effectiveness. However, these median points would vary from sequences to sequences and a set of fixed search points cannot take much benefit from it. We therefore propose the pseudo statistical median search points to replace the original fixed median points to achieve better search efficiency.

C. Double Layered Search Pattern Concept

In summary, the proposed double-layered search pattern has following characteristics. 1) The necessary search point numbers for small motion cases should be minimized. 2) Search points of large motion search points will be checked only if necessary. 3) To be able to catch uncommon large motion with effective 2–D logarithmic method. To take advantage of the motion characteristics, an adaptive double-layered initial search pattern is proposed in the following. It uses iterative SDSP/E-HEXBS as a convergence method during later search steps.

III. Proposed Algorithm

In the section, we propose an adaptive double-layered initial search pattern (ADLISP). This algorithm has two phases for each macroblock process: motion estimation and adaptive adjustment of search pattern.

The initial search step of motion estimation phase consists of a pattern with eight search points, four in both the inner and outer layers. The inner layer is a small diamond search pattern. The outer layer is a large cross pattern, which is consist of pseudo statistical median points. During later steps of motion estimation phase, two different converge algorithms, iterative SDSP and E-HEXBS, are used for different situations. The outer layer search points are then adjusted after the motion estimation phase.

A. Motion Estimation

At the beginning of the motion estimation phase of each macroblock, we use the MPEG-4 method to predict motion vectors from the median of the three neighboring macroblocks. The predicted motion vector and the nonmoving (0, 0) vector are checked first. The vector with the minimum distortion is used as the center of the search range. After the search center is set, the initial search pattern is applied.

The four inner layer points, (± 1, 0) and (0, ± 1), are then checked. If the minimum BDM vector still falls in the search center, the search is completed and has used six or five [if the

[3]Averaged value of absolute weight of motion vectors over single axis. For example, the average movement of vectors (+3, −2), (+1, −2), and (−2, +5) are 2 in horizontal and 3 in vertical.

TABLE III
VISUAL QUALITY COMPARISON OF TEST RESULTS

		Full	ADLISP		CDHS-T		DCS		DS		E3SS		E-HEXBS	
		MSE	MSE	MSEI	MSE	MSEI	MSE	MSEI	MSE	MSEI	MSE	MSEI	MSE	MSEI
CIF	Akiyo	4	4	1.0%	4	1.1%	4	2.2%	4	0.5%	4	1.0%	6	55.4%
	Basket Ball	193	201	4.0%	200	3.6%	205	6.3%	198	2.4%	198	2.4%	307	59.3%
	Bus	210	299	42.2%	352	67.3%	362	72.4%	331	57.5%	300	42.9%	416	97.8%
	Canoa	231	256	10.6%	258	11.8%	264	14.4%	253	9.4%	247	6.9%	297	28.5%
	Coastguard	62	63	1.4%	63	1.1%	65	4.1%	63	0.9%	62	0.5%	79	27.5%
	Dancer	53	62	15.8%	61	15.0%	65	21.0%	60	12.3%	59	11.1%	72	34.6%
	Football	211	262	24.5%	262	24.3%	279	32.4%	252	19.8%	239	13.6%	288	36.6%
	Foreman	47	53	13.1%	53	12.4%	55	17.7%	51	9.3%	50	7.3%	71	51.7%
	Hall Monitor	22	23	2.8%	23	2.6%	23	4.5%	23	1.5%	23	1.8%	25	13.2%
	Kiel	236	248	5.1%	248	5.1%	251	6.2%	245	3.9%	245	3.7%	502	112.5%
	Mobile	232	237	2.2%	237	2.1%	238	2.7%	236	1.5%	236	1.7%	477	105.5%
	Silent	19	22	19.9%	22	20.0%	23	24.7%	21	14.9%	21	10.1%	25	32.2%
	Singer	15	16	7.6%	16	6.8%	17	13.0%	16	2.8%	16	6.2%	22	47.3%
	Stefan	305	352	15.4%	378	23.7%	388	27.1%	371	21.6%	337	10.3%	465	52.5%
	Table Tennis	60	77	29.2%	76	28.1%	80	34.7%	73	21.6%	71	18.4%	96	61.5%
	Tempete	137	148	7.8%	148	8.1%	149	8.5%	147	6.8%	142	3.3%	177	28.7%
	Vectra	184	204	11.1%	231	25.6%	239	30.3%	228	24.1%	199	8.1%	263	43.4%
QCIF	Carphone	64	72	12.3%	72	13.2%	74	15.6%	70	10.0%	69	7.7%	86	34.4%
	Coastguard	42	43	2.4%	43	2.2%	43	2.7%	43	2.0%	42	0.9%	95	125.9%
	Foreman	55	62	13.4%	61	12.8%	63	15.4%	59	7.6%	58	6.8%	77	41.6%
	Hall Monitor	15	16	0.7%	16	0.6%	16	1.5%	15	0.1%	16	0.6%	17	12.3%
	Mobile	166	166	0.1%	166	0.1%	166	0.1%	166	0.1%	166	0.1%	179	7.8%
	Stefan	257	284	10.3%	343	33.3%	342	33.0%	326	26.8%	274	6.7%	423	64.5%
	Table Tennis	79	91	15.1%	92	16.0%	97	22.9%	87	10.2%	87	10.2%	119	49.5%
Average		**121**	**136**	**11.2%**	**143**	**14.0%**	**146**	**17.2%**	**139**	**11.2%**	**132**	**7.6%**	**191**	**51.0%**

predicted motion vector is (0, 0)] block-matching calculations. Otherwise, the four points of the outer layer are then checked. If the renewed minimum BDM vector falls on the inner layer SDSP, then the search continues with normal iterative SDSP search until an optimal match is found. Otherwise, if the minimum BDM vector falls on the outer layer, then the following several processes are performed.

The outer layer search points of the initial step are ($\pm \boldsymbol{r_x}$, 0), and (0, $\pm \boldsymbol{r_y}$), which form a large asymmetric cross pattern. $\boldsymbol{r_x}$ and $\boldsymbol{r_y}$ are adjustable and are initially given as $\boldsymbol{r_x} = 5$ and $\boldsymbol{r_y} = 5$ according to the common average movement observed in last section. If the best vector of the initial search falls on the outer layer, then the search continues with an E-HEXBS search on later steps.

During the iterative SDSP search, the small diamond checkpoints centered on the best-matched search point are checked. After block matching calculation, if any of those search point has even lower distortion, another set of SDSP centered on the newly found best position are then applied. The process repeats until the best-matched search point no longer changes.

E-HEXBS is similar to iterative SDSP except its check pattern is a six-point hexagon. Because the six-side-based fast inner search [6] is used in E-HEXBS, only two or three more search points are used after applying E-HEXBS to determine the final motion vector. Note that only the horizontal hexagon search pattern as shown in Fig. 1(c) is used because of the higher horizontal weight of most motion vectors.

B. Adjustment of Search Pattern

Whenever a search is complete, the estimated vector $(\boldsymbol{v_x}, \boldsymbol{v_y})$ is used to adjust the outer layer to a more suitable position, which is ideally equal to the statistical median point. Because real-time movement calculation is extensive, four outer layer pseudo median points derived from the rules below are then used.

The outrange indicators $\boldsymbol{C_x}$ and $\boldsymbol{C_y}$ are set to zero initially before the process of entire video sequence. $\boldsymbol{T}$ is a threshold to determine outer layer's sensitivity with recent motion statistics.

1) If the best vector of the initial search step is NOT located on the outer layer,
   goto step 7.
2) If $|\boldsymbol{v_x}| + |\boldsymbol{v_y}| < 4$,
   goto step 7.
3) If $|\boldsymbol{v_x}| > \boldsymbol{r_x}$,
   the horizontal outrange indicator $\boldsymbol{C_x} := \boldsymbol{C_x} + 1$.
   else if $|\boldsymbol{v_x}| < \boldsymbol{r_x}$,
   $\boldsymbol{C_x} := \boldsymbol{C_x} - 1$..
4) If $|\boldsymbol{v_y}| > \boldsymbol{r_y}$,
   the vertical outrange indicator $\boldsymbol{C_y} := \boldsymbol{C_y} + 1$.
   else if $|\boldsymbol{v_y}| < \boldsymbol{r_y}$,
   $\boldsymbol{C_y} := \boldsymbol{C_y} - 1$..
5) If $\boldsymbol{C_x} > +\boldsymbol{T}$,
   $\boldsymbol{r_x} := \boldsymbol{r_x} + 1; \boldsymbol{C_x} := 0$.
   else if $\boldsymbol{C_x} < -\boldsymbol{T}$,
   $\boldsymbol{r_x} := \boldsymbol{r_x} - 1; \boldsymbol{C_x} := 0$.
6) If $\boldsymbol{C_y} > +\boldsymbol{T}$,
   $\boldsymbol{r_y} := \boldsymbol{r_y} + 1; \boldsymbol{C_y} := 0$.
   else if $\boldsymbol{C_y} < -\boldsymbol{T}$,
   $\boldsymbol{r_y} := \boldsymbol{r_y} - 1; \boldsymbol{C_y} := 0$.
7) End of adaptive adjustment.

TABLE IV
SPEED COMPARISON OF TEST RESULTS

		Full	ADLISP		CDHS-T		DCS		DS		E3SS		E-HEXBS	
		SP	SP	SU	SP	SU	SP	SU	SP	SU	SP	SU	SP	SU
CIF	Akiyo	868.3	4.9	177	4.9	177	4.9	177	12.2	71	12.3	71	8.8	98
	Basket Ball	824.7	7.0	118	7.3	113	6.8	121	13.4	61	13.9	59	9.5	87
	Bus	746.7	8.0	94	8.9	83	8.2	91	14.9	50	16.0	47	10.7	70
	Canoa	661.6	9.3	71	10.7	62	9.3	71	15.8	42	15.8	42	11.2	59
	Coastguard	819.4	6.5	126	6.5	126	6.4	128	12.7	64	13.5	61	8.8	93
	Dancer	632.4	10.6	60	12.4	51	10.4	61	17.1	37	16.3	39	11.7	54
	Football	646.3	12.2	53	13.1	49	11.1	58	18.1	36	18.4	35	12.3	53
	Foreman	767.9	7.9	97	8.1	94	7.4	103	14.1	54	14.3	54	9.9	77
	Hall Monitor	861.6	5.8	148	6.1	142	5.7	150	12.8	67	13.1	66	8.9	97
	Kiel	776.4	7.4	104	7.4	105	7.2	108	13.8	56	14.3	54	10.5	74
	Mobile	845.6	5.6	150	5.6	152	5.5	155	12.4	68	12.7	67	8.8	96
	Silent	853.4	6.1	141	6.1	139	5.8	147	13.1	65	13.3	64	9.2	93
	Singer	868.2	5.1	170	5.1	170	5.0	172	12.3	71	12.4	70	8.6	101
	Stefan	737.7	8.1	91	9.4	79	8.3	89	15.2	49	15.9	46	10.7	69
	Table Tennis	839.3	6.5	129	6.9	122	6.4	132	13.3	63	14.0	60	9.4	89
	Tempete	854.5	6.5	131	6.6	130	6.2	137	13.1	65	13.6	63	9.1	94
	Vectra	592.6	8.1	73	10.3	57	9.3	64	16.1	37	14.3	41	10.8	55
QCIF	Carphone	766.6	6.6	117	7.0	109	6.4	119	12.8	60	13.3	58	8.9	87
	Coastguard	761.4	5.4	141	5.3	143	5.2	146	11.7	65	11.9	64	8.3	92
	Foreman	737.6	6.8	109	6.9	108	6.3	116	12.7	58	13.0	57	8.9	83
	Hall Monitor	781.6	5.0	158	4.9	159	4.9	160	11.5	68	11.6	67	8.0	97
	Mobile	780.1	4.8	163	4.7	164	4.7	165	11.4	68	11.5	68	7.9	99
	Stefan	706.7	7.1	100	8.0	89	7.2	98	13.5	52	13.6	52	9.3	76
	Table Tennis	770.3	6.0	129	6.2	124	5.8	132	12.3	63	12.6	61	8.6	89
Average		**770.9**	**7.0**	**119**	**7.4**	**114**	**6.9**	**121**	**13.6**	**58**	**13.8**	**57**	**9.5**	**83**

SU: Speed up relative to full search

With this feedback function, $\boldsymbol{r_x}$ and $\boldsymbol{r_y}$ are adjusted to approach the local statistical median according to recent estimated motion vectors and therefore can be used to decide pseudo statistical median search points ($\pm\boldsymbol{r_x}$, 0), and (0, $\pm\boldsymbol{r_y}$).

Real video sequences generally contain a mixture of motion types. For example, some frames are quasi-statistic while others have large inconsistent motion which make the statistical median varies from time to time. With a smaller threshold $\boldsymbol{T}$, the outer layer search points are more sensitive to recent motion vectors behavior, but also become more easily to be affected by noise. After several tests, we found that setting the threshold $\boldsymbol{T}$ to 32 achieved the best results. This value is then used empirically.

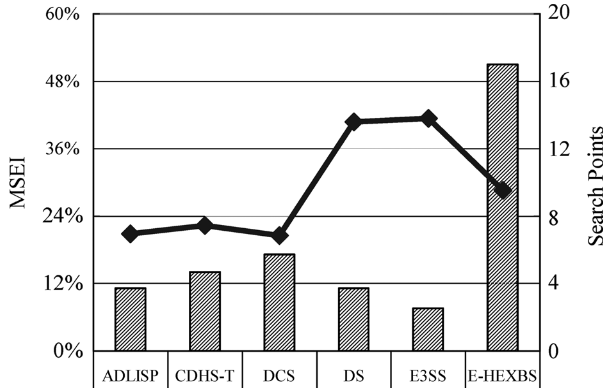
The system diagram of the proposed algorithm is shown in Fig. 6. The examples of search points using ADLISP are shown in Fig. 7. Fig. 7(a) gives an example that the estimated motion vector locates at the center of search range. After checking (0, 0), ($\pm$1, 0) and (0, $\pm$1), the best-match vector locates in the center. Therefore the outer layer search points are skipped and the search ends with five block-matching calculations. In Fig. 7(b) the best vector of initial search falls on the inner layer of ADLISP after checking all nine initial search points, and therefore iterative SDSP is applied. After first SDSP search, the best-matched vector moves from (0, 1), which is selected after initial search step, to (1, 1). Second SDSP search from (1, 1) does not find any better vector and therefore the search is done with the output motion vector (1, 1).

In Fig. 7(c) the best vector of initial search falls on (0, 4), which is one of the outer layer search point. E-HEXBS is then applied from (0, 4). The best-matched vector moves from (0, 4) to $(-2, 4)$ after first hexagon pattern search. The second hexagon pattern search from ($-$2, 4) does not find any better vector on the six checkpoints, therefore the search continues with 6-side-based fast inner search. In this example, ($-$4, 4) and ($-$3, 6) have minimum distortion among the six search points of second hexagon pattern. According to the rule of six-side-based fast inner search, $(-3, 4)$ and $(-3, 5)$ need to be further checked. The final result is the motion vector $(-3, 5)$ that has lowest distortion of all searched vectors.

IV. Simulation Results

To evaluate and compare performance of the proposed algorithm, motion estimation performance is tested experimentally for the proposed algorithm and five other algorithms: DS, CDHS-T, E-HEXBS, E3SS and DCS. The frame structure is IPPP, and the search range is set to 15, i.e., the search window size is 31 by 31. Only the luminance component values of the video sequences are used. Although ADCS gains superior search speed from its early termination judgment, no early termination is used for comparison on search pattern effectiveness in these experiments. All video sequences listed in Table I are tested.

Simulation results are expressed in the number of required search points (SP) and mean square error (MSE). Table III lists the visual quality comparison of the results. Search speed is listed in Table IV. Average experimental results of all sequences are charted in Fig. 8. For easier comparison, these algorithms are all compared with the result of the full search results, which gives speedup and MSE increase (MSEI) relative to full search for each of the tested algorithms. This is convenient because the



	ADLISP	CDHS-T	DCS	DS	E3SS	E-HEXBS
MSEI	11.2%	14.0%	17.2%	11.2%	7.6%	51.0%
Search Points	6.96	7.44	6.86	13.60	13.81	9.54

Fig. 8. MSEI and search points simulation results.

results of the tested algorithms can be represented in percentage. A higher MSEI means a lower visual quality.

In comparison of video algorithms, E3SS performs the best quality and ADLISP and DS come after. CDHS-T and DCS has little more quality loss and E-HEXBS performs relatively worst.

In terms of computation cost, DCS, ADLISP and CDHS-T spend least overall search points. They are fast and have only little speed difference. Slower algorithms DS and E3SS spend twice number of block-matching calculation than the three fast methodologies above while E-HEXBS has moderate speed.

On the close observation of the results sequence by sequence, ADLISP usually spends less search points but generates relatively lower MSEI on sequences with inconsistent motion, such as sequences Foreman and Stefan. It shows that ADLISP generally surpasses other algorithms in the tests with inconsistent motion cases, which usually are bottlenecks of most fast motion estimation algorithms. In the cases with smooth motion, ADLISP is only slightly inferior to CDHS-T and DCS.

Overall, ADLISP has superior performance on both speed and visual quality. It has additional advantage on inconsistent motion video environments.

V. Conclusion

This paper has proposed the ADLISP adaptive double layered initial search pattern for fast motion estimation. By using an adaptive size for the initial cross pattern, both large and small motion can be easily estimated. The solution provides high performance for video sequences with widely varying motion characteristics. Experimental results show that ADLISP outperforms other search patterns not only in terms of speed but also in quality. ADLISP offers visual quality comparable with E3SS yet costs as few search points as DCS and CDHS-T. Because of using pseudo-statistical median, ADLISP benefits from effective 2-D logarithmic without taking high statistic computation. As a summary, ADLISP is a cost-effective and easy-implement motion estimation algorithm in video coding systems, especially in high motion environments.

References

[1] *Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s—Part 2: Video*, JTC1/SC29/WG11, ISO/IEC 11172-2 (MPEG-1 Video), 1993.

[2] *Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video*, ITU-T and ISO/IEC JTC 1, ITU Rec. H.262—ISO/IEC 13818-2 (MPEG-2 Video), 1994.

[3] *Information Technology—Coding of Audio Visual Objects—Part 2: Visual*, JTC1/SC29/WG11, ISO/IEC 14496-2 (MPEG-4 Visual), 2000.

[4] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

[5] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 349–355, May 2002.

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

[7] X.-Q. Banh and Y.-P. Tan, "Adaptive dual-cross search algorithm for block-matching motion estimation," *IEEE Trans. Consumer Electron.*, vol. 50, no. 2, pp. 766–775, May 2004.

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

[9] X. Jing and L.-P. Chau, "An efficient three-step search algorithm for block motion estimation," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 435–438, Jun. 2004.

[10] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799–1808, Dec. 1981.

[11] A. Puri, H. M. Hang, and D. L. Schilling, "An efficient block matching algorithm for motion compensated coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1987, pp. 1063–1066.

[12] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419–423, Aug. 1996.

[13] J. Lu and M. L. Liou, "A simple and efficient search algorithm for block matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 429–433, Apr. 1997.

[14] CCITT SGXV, Description of Reference Model 8 (RM8) Document 525, Working Party XV/4, Specialists Group on Coding for Visual Technology, Jun. 1989.

[15] S. C. Tai, C. S. Yu, and C. W. Lee, "Double-layered initial search pattern for fast motion estimation," in *Proc. IPSI-2005*, Amalfi, Italy, Feb. 2005.

Chong-Shou Yu received the B.S. and M.S. degrees in electrical engineering from the National Cheng-Kung University (NCKU), Tainan, Taiwan, R.O.C., in 1998 and 2000, respectively. He is currently pursuing the Ph.D. in electrical engineering at NCKU. His research interests include video coding, data compression and digital watermark.

Shen-Chuan Tai (M'96–SM'05) received the B.S. and M.S. degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1982 and 1986, respectively, and the Ph.D. degree in computer science from the National Tsing Hua University, Hsinchu, Taiwan, in 1989.

He is currently a Professor of Electrical Engineering in the Department of Electrical Engineering, National Cheng-Kung University, Tainan, Taiwan. His teaching and research interests include data compression, DSP VLSI array processor, computerized electrocardiogram processing, multimedia system, and algorithms.