A Novel Kite-Cross-Diamond Search Algorithm For Fast Video Coding and Videoconferencing Applications

Chi-Wai Lam, Lai-Man Po and Chun Ho Cheung
Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR
Email: cwlam@ee.cityu.edu.hk, eelmpo@cityu.edu.hk, terence@ieee.org

ABSTRACT

In this paper, we propose a kite-cross-diamond search (KCDS) algorithm, which is an improved version of the well-known cross-diamond search (CDS) algorithm and small cross-diamond search (SCDS) algorithm. Unlike traditional search pattern in block matching algorithm, such as square, diamond or cross – all are in vertically and horizontally symmetric shape, the KCDS algorithm adopts a novel asymmetric kite-shaped search patterns in the search step to keep similar distortion while the speed of the motion estimation for stationary or quasi-stationary blocks is further boosted. Experimental results show that the KCDS algorithm could achieve 39% searching point reduction as compared with CDS whereas similar and even better prediction accuracy is resulted in low-motion sequences. Simulations show that KCDS is the fastest algorithm and it performs more accurate in some kinds of sequences. This algorithm is especially suitable for videoconferencing applications.

Index terms—Motion estimation, kite-cross-diamond search, cross-center biased characteristic.

1. INTRODUCTION

Motion estimation (ME) is a process to estimate the pels or pixels of the current frame from reference frame(s). Block matching algorithm (BMA), which is a temporal redundancy removal technique between 2 or more successive frames, is an integral part for most of the motion-compensated video coding standards. Frames are being divided into regular sized blocks, or so-called macroblocks (MB). Block-matching method is to seek for the best-matched block from the previous frame, usually the first single frame, within a fixed-sized of search window (*w*). Based on a block distortion measure (BDM) or other matching criteria, the displacement of the best-matched block will be described as the motion vector (MV) to the block in the current frame. The best match is usually evaluated by a cost function based on a BDM such as Mean Square Error (MSE), Mean Absolute Error (MAE) or Sum of Absolute Differences (SAD). Full search (FS) method, which performs searching all the candidate blocks within the search window exhaustively, introduces high intensive computation. Over the last two decades, many fast motion estimation algorithms has been proposed to give a faster estimation with similar block distortion compared to FS. Some well known fast BMA are the three-step search (3SS) [1], the new three-step search (N3SS) [2], the four-step search (4SS) [3], the diamond search (DS) [4], the cross-diamond search (CDS) [5] and small cross-diamond search (SCDS) [6]. As the characteristic of center-biased MV distribution (MVD) which inspired many fast BMA in last decade, more than 80% of the blocks can be regarded as stationary or quasi-stationary blocks, i.e. most of the motion vectors are enclosed in the central 5x5 (blocks) area. This center-based characteristic can be even found in the fast-motion sequences. To exploits this phenomenon, NTSS added 8 center-neighboring blocks and introduced a halfway-stop technique to achieve crucial speedup for stationary and quasi-stationary blocks. 4SS also exploits the center-biased properties of motion vectors distribution by using halfway-stop techniques and smaller square search pattern compared to 3SS. DS was proposed with two novel ideas: a diamond shape searching pattern and unrestricted searching steps. DS is a highly center biased by using a compact diamond search pattern, and the unrestricted searching steps is used for reducing the chances of being trapped by local optima. Recently, CDS and it's improved variant SCDS algorithm exploit a more dominant cross-center-biased (CCB) property in most real-world sequences. These two algorithms not only maintain similar distortion error, but also outperform other fast BMA by their speed performance. In this paper, a novel fast BMA called kite-cross-diamond search algorithm (KCDS), which is an improved version of the CDS algorithm and SCDS algorithm, is proposed. Similar to SCDS, it uses a small cross-shaped search patterns in the first step and results in higher speed for the motion estimation of stationary block. Similar starting pattern can be found in [7]. Then it uses a kite-shape pattern in second step and a so-called biased-corner pattern in third step to improve the accuracy in searching for quasi-stationary blocks. Experimental simulations show that it can achieve fewer search points over CDS and SCDS, and can obtain the similar distortion performance. The result also shows that it is favorable in videoconferencing sequences. This paper is organized as follows. The second section introduces the CCB MVD property. The third section presents the details of the kite-cross-diamond search algorithm. The fourth section describes the experimental result and some performance evaluations. Some concluding remarks are given in the last section.

2. CROSS CENTER-BIASED MVP DISTRIBUTION

Frame Format (Numbers of frames)	*Sequences*
CIF (352×288, 80 frames)	Miss America, Sales, Claire
SIF (352×240, 80 frames)	Tennis, Garden, Football

TABLE I: Video Sequences used for Analysis

There is a well-known property of image sequences - *The block motion field of a real world image sequence is usually gentle, smooth, and varies slowly*. To demonstrate the property of the global minimum motion vector distribution, by applying FS with spiral block-matching style and MAD as the BDM on the six well-known real-world image sequences, which is listed in Table

Probablities (%) at corresponding checking-point within the search window

r_Ver \ r_Hor	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
7	0.062	0.02862	0.04252	0.02662	0.04	0.0543	0.03968	0.37227	0.03588	0.0441	0.02432	0.01885	0.03295	0.01768	0.048
6	0.013	0.02327	0.00622	0.01483	0.0122	0.02285	0.04115	0.11553	0.05008	0.01517	0.01527	0.00968	0.01053	0.0201	0.013
5	0.006	0.00652	0.00517	0.00358	0.01188	0.0239	0.0161	0.11638	0.05063	0.01348	0.00517	0.0061	0.00412	0.0039	0.006
4	0.033	0.05345	0.0181	0.0342	0.05188	0.07123	0.11227	0.18973	0.18613	0.04883	0.02842	0.02683	0.01853	0.03798	0.019
3	0.051	0.02778	0.04967	0.03033	0.09733	0.25547	0.34103	0.77127	0.35733	0.17993	0.06618	0.02578	0.03895	0.02012	0.047
2	0.015	0.02578	0.00643	0.02452	0.06072	0.25075	0.69938	**0.624**	0.41003	0.1612	0.03422	0.02737	0.01748	0.01957	0.023
1	0.028	0.01832	0.02442	0.05295	0.1287	0.41457	**0.738**	**4.216**	**0.623**	0.2538	0.08102	0.04705	0.03682	0.0283	0.04
0	0.161	0.36468	0.09355	0.22695	0.39983	**0.814**	**4.378**	**45.49**	**3.44**	**0.516**	0.30798	0.22548	0.11332	0.25085	0.131
-1	0.028	0.00758	0.02525	0.04832	0.11068	0.39058	**0.757**	**11.49**	**0.664**	0.34932	0.10743	0.05103	0.03093	0.01305	0.029
-2	0.015	0.03313	0.01212	0.04733	0.07207	0.18257	0.38333	**3.776**	0.52453	0.24832	0.05852	0.03872	0.01843	0.02008	0.016
-3	0.045	0.02115	0.04695	0.0324	0.09555	0.25347	0.43328	2.95938	0.23842	0.23212	0.05493	0.03313	0.03808	0.02758	0.048
-4	0.015	0.03935	0.0118	0.02958	0.05093	0.07218	0.45172	0.67857	0.1452	0.10112	0.05567	0.03378	0.0181	0.04398	0.013
-5	0.008	0.00547	0.00768	0.0098	0.01338	0.02935	0.64438	0.57385	0.05008	0.04937	0.01917	0.01935	0.01273	0.00812	0.01
-6	0.024	0.02063	0.01453	0.0159	0.01537	0.022	0.17688	0.29398	0.03218	0.03693	0.02747	0.02462	0.01525	0.02598	0.011
-7	0.06	0.01463	0.036	0.01557	0.04535	0.04715	0.0524	0.35742	0.06335	0.06788	0.06135	0.06377	0.05252	0.02537	0.068

TABLE II: Average distribution measured at distance r using 6 CIF/SIF sequences for $|w| = 7$

I. The average MV probabilities (MVP) distributions is tabulated in Table II. The CIF sequences can be regarded as low motion (video conference) video, including "Miss America", "Salesman", and "Claire". These sequences is relatively gentle, smooth, and with low-motion content. Whereas another three SIF video sequences "Football", "Garden" and "Tennis" are relatively with high motion content. Zooming, fast movement object, and panning can be found in these 3 sequences. From observing the MVP distributions on different sequences, we found that most real-world sequences possess the center-biased MVD characteristic (over 80% MVP of the blocks having motion vectors within central 5×5 grid or radius-$r = \pm 2$), instead of an uniform distribution. The result also shows that the *cross-center-biased MV distribution* is more dominant within this radius. For instance, in Fig.1, 71.76% of the motion vectors are found located in the central 2×2 area, i.e., A+B+D or $r = \pm 1$. And there is about 68.98% of motion vector are located in A+B or the cross-center region. Moreover, to look at 5 × 5 area (A+B+C+D+E), total MVP is 81.75% and the cross-center probabilities within this area (A+B+C) has accounted for 74.71%. Furthermore, the probabilities sum of the 4 points in E (position within the cross) is higher than D (diagonal position). By observing the above analysis, we conclude that the cross-center distribution dominates the corresponding square region. Inspired from such a highly cross-center based distribution, the searching pattern of BMA in first few steps can be matched the cross-center shape to save the searching point for stationary and quasi-stationary while maintain a similar distortion error. Another observation from the MVD is the effect of the gravity - the vertical MVP down below the center point is holding a significant probability. e.g. the probability of $r\_Ver = -3$, $r\_Hor = 0$ is almost 3% whereas the opposite position in upper ($r\_Ver = 3$, $r\_Hor = 0$) is only ~0.8%.

3. KITE-CROSS-DIAMOND SEARCH (KCDS) ALGORITHM

In this section we first to describe the search patterns used in the algorithm, and later the search path strategy will be explained.

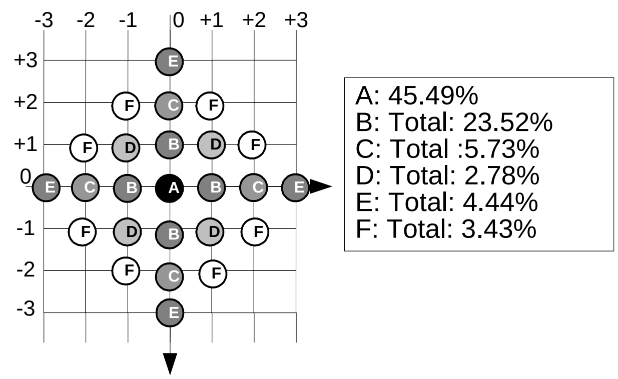

Fig. 1. The MVD within r=4 diamond shape area

A. *Search Patterns*

The search-point configuration used in the KCDS is divided in 4 different shapes: cross-shaped pattern, diamond-shaped pattern, kite-shaped pattern (KSP) and biased-corner pattern (BCP). Fig.2 (a) show the small cross-shaped pattern (SCSP) and the large cross-shaped pattern (LCSP). The same search pattern from DS: small diamond-shaped pattern (SDSP) and large diamond-shaped pattern (LDSP) are shown in Fig.2 (b). Unlike traditional search pattern, such as square, diamond, cross – all are vertically and horizontally symmetry, in the kite-shaped pattern (KSP), only the diagonal that connects the longer ends of the kite is the line of symmetry. Fig 3 (a) shows the vertical-kite by described as up-kite in which the dart (the most outer vertex) is pointing up. Fig 3 (b) is called left-kite. For another 2 KSP –down-kite and right-kite are shown in Fig 3 (c) and (d). BCP, also is shown in Fig.3, is sharing the same center of the KSP and it depends on the direction of the dart to indicate the biased point of searching.

B. *The KCDS algorihtm*

From the simulation result on those six well-known sequences, we found that nearly 70% blocks that can be regarded as stationary ($r = 0$) or quasi-stationary blocks ($r = 1$). For the sake of this highly small cross-center-biased property in most real world sequences, we take the cross-shaped patterns as first step to the KCDS. The difference between KCDS and CDS is that the first step of KCDS is a SCSP, which is saving the number of search point for stationary. And the difference between KCDS and SCDS is that the KSP and BCP are employed in consequence steps to improve the accuracy for quasi-stationary blocks by taking few more significant point (point E and F in the direction [Fig. 1]) in the search pattern. The details and the analysis of the algorithm are given below:

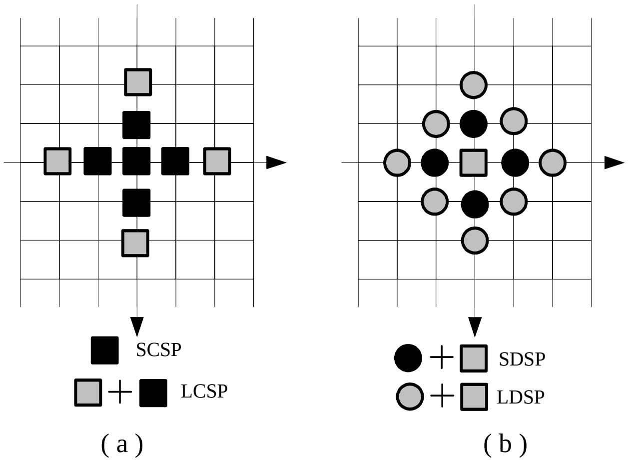

Fig. 2 Search Patterns used in the kite-cross-diamond search

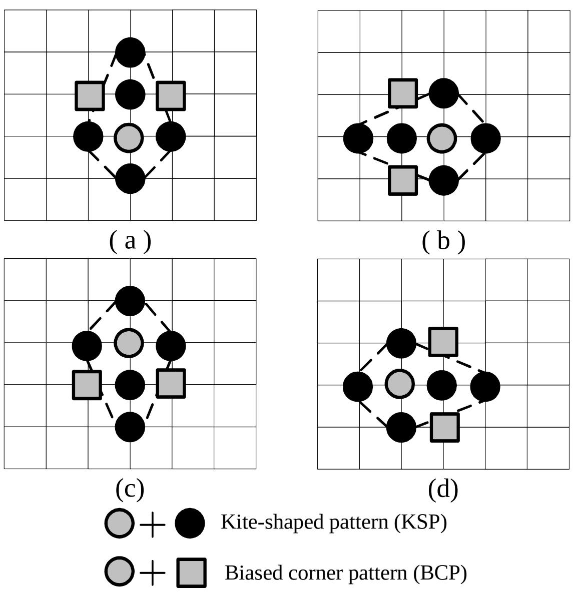

Fig. 3. Kite Search Patterns and Biased Corner patterns: With vertical symmetry (a) up-kite. and up-biased corner(c) down-kite and down-biased corner; with horizontal symmetry: (b) left-kite and left-biased corner (d) right-kite and right-biased corner.

Step 1 *(Starting - SCSP):* A minimum BDM is found from the 5 search points of the SCSP [Fig.2 (a)] located at the center of search window. If the minimum BDM point occurs at the center of the SCSP (0,0), the search stops (First Step Stop); otherwise, go to Step 2.

Step 2 *(KSP):* With the vertex (minimum BDM point) in the first SCSP as the center, a particular KSP is formed based on the motion direction in previous step. For example, if the minimum BDM is located in upper vertex in first step, the new KSP will be an up-kite shape (the dart is pointing up) described as Fig 3 (a). Thus, depends on the MV direction in step 1, there are 4 cases of newly formed KSP in this step: up-kite, down-kite, right-kite and left-kite, If the minimum BDM point occurs at the center of this KSP, then go to Step 3; otherwise go to Step 4.

Step 3 *(BSP):* Checking the two BCP points by following the biased of KCP of previous step. For example, if there is a up-kite in previous step, the BCP will be the up-biased corner against the center. If the minimum BDM point is still unchanged, then the search stop (third-step stop, e.g. Fig.4(b)). Otherwise go to Step 4.

Step 4 *(Diamond Searching)*: A new Large-Diamond-Shaped Pattern LDSP is formed by repositioning the minimum BDM found in previous step as the center of the LDSP. If the new minimum BDM point is at the center of the newly formed LDSP, then go to Step 5 for converging the final solution; otherwise, this step is repeated.

Step 5 *(Ending):* With the minimum BDM point in the previous step as the center, a SDSP is formed. Identify the new minimum BDM point from the SDSP, which is the final solution.

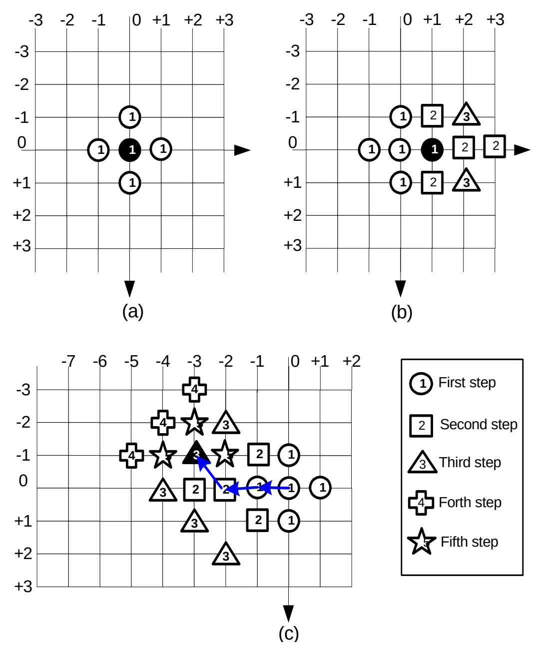

Fig. 4.Examples of the KCDS: (a) first-step-stop with MV(0,0). (b) Third-step-stop with MV(1,0). (c) an unrestricted search path for MV(-3,-1) .

C. Analysis of KCDS algorithm

KCDS is regarded as an improved version of CDS and SCDS because both of them also focus on advancing the speed and the quality performance of videoconferencing sequences. Moreover, they also employ the cross-shape pattern in the first step. To compare the CDS, the main improvement of this algorithm is the speed performance; KCDS reduces the number of search points significantly if there is a stationary or quasi-stationary block. The configuration of the searching patterns aims to fit the small CCB MVD characteristics. Thus, it provides more chance to save up the searching points for motion vectors. In order to meet the tendency of motion vector, the kite shape pattern and the biased corner pattern in following steps improve the quality by searching the points on *E* and *F* point shown in Fig 1. In Fig 4, it shows 3 typical examples of KCDS and each candidate point is marked with the corresponding step number. Fig.4 (a) and (b) show the two halfway-stop examples. As same as SCDS, the KCDS takes 5 (first step stop) and 11 (third step stop), whereas the CDS took 9 and 11 search points, and the DS took 13 search points for searching the same block respectively. Although KCDS takes the same number of searching point with SCDS in the halfway-stop cases, KCDS is more accurate to quasi-stationary block because the searching pattern attempts to match the tendency of the motion vector. Another search paths for $r > 1$ are shown in Fig 4 (c). Start from step 4, the subsequent steps will be exactly the same as diamond search.

4. EXPERIMENTAL RESULTS

In our simulations, the mean absolute error (MAE) used as the BDM. The block size is at 16×16, and the maximum displacement in the search areas is ±7 pixels in both the horizontal and the vertical directions. The simulation is performed with six sequences with different degrees and types of motion content as listed in Table I. We compared the KCDS against CDS and SCDS using the following test criteria: 1) Average searching point (ASP) – the average number of point used to find the motion vector; and 2) Average MAE per pixel– This shows the magnitude of distortion per pixel. Table III and IV summarize the experimental results of each search strategy over the test criteria using the tested sequences. And the speed/MAD improvement in percentage of the KCDS over CDS and SCDS are tabulated in Table V. By observing the result, The KCDS takes the smallest average number of search points per block among other fast BMA for all tested sequences -. To compared with CDS, among the video conferencing sequence, such as "Miss America", "Sales", and "Claire", the proposed KCDS obtains at most ~39% of speed improvement - *% of point reduction*, even in vigorous motion content like "Football", the speed up ratio can being achieved up to ~27% and the least improvement is ~8%. The trade off of the block distortion for faster speed is tabulate in Table IV which compare the difference of average MAE per pixel from FS. The result shows that the KCDS gives nearly the same MAE performance as compared to CDS and SCDS in most sequences. In the videoconferencing sequences KCDS even perform better, although the quality improvement is small (<0.2%). For high motion content, the KCDS introduce slight quality degradation as compared to CDS and SCDS (maximum ~4% of the degradation in "Tennis" with the trade off of at least 15% speed improvement). Therefore, KCDS is more robust, in which this is the fastest among all BMAs and more accurate compared to CDS and SCDS in all tested video conferencing sequences. For high motion sequence, it still maintains a satisfying tradeoff between error distortion and speedup ratio.

5. CONCLUSION

By observing the cross-center biased motion vector distribution characteristics of the real world video sequences, we proposed a kite-cross-diamond search (KCDS) algorithm, which emphasis a novel idea of kite shape pattern. Simulation results showed that KCDS improve high degree of speedup ratio while providing similar or even better prediction accuracy. It is especially suitable for videoconferencing application.

ACKNOWLEDGMENT

The work described in this paper was substantially supported by a grant from City University of Hong Kong, Hong Kong SAR, China. [Project No.7001385]

Average searching point ASP

	FS	3SS	4SS	N3SS	DS	CDS	SCDS	**KCDS**
Tennis	202.1	23.20	18.65	20.67	16.25	15.38	13.9	**13**
Garden	202.1	23.24	18.80	21.38	16.84	15.09	14.87	**13.82**
Football	202.1	23.06	16.69	17.65	13.67	10.96	8.24	**7.99**
MissA	202.1	23.46	18.319	19.99	16.36	11.75	10.75	**10.69**
Claire	202.1	23.22	15.924	16.19	12.4	8.92	5.38	**5.36**
Sales	202.1	23.21	16.206	16.94	13.02	9.5	6.99	**6.98**

TABLE III: The Average number of searching points of FS,3SS, 4SS, N3SS, DS, CDS, SCDS, and KCDS over the six sequences

Difference of average MAE per pixel from FS

	3SS	4SS	N3SS	DS	CDS	SCDS	**KCDS**
Tennis	1.0374	0.4383	0.488	0.2415	0.2935	0.3584	**0.5281**
Garden	0.9845	0.6502	0.1568	0.2337	0.1906	0.2056	**0.2159**
Football	0.2436	0.1683	0.1034	0.1452	0.1709	0.19	**0.2142**
MissA	0.1169	0.1165	0.0253	0.1021	0.0352	0.0371	**0.0328**
Claire	0.0038	0.0035	0.001	0.0014	0.0029	0.0033	**0.0028**
Sales	0.0521	0.044	0.0081	0.0423	0.0094	0.01	**0.0084**

TABLE IV: Differences of average MAE per pixel from FS $(MAE_{FS} - MAE_{3SS, 4SS, N3SS, DS, CDS, SCDS, and KCDS})$ over the six sequences.

	KCDS over CDS		**KCDS over SCDS**	
	SIR (%)	MAE	SIR (%)	MAE
Tennis	**-15.4047**	4.101299	**-6.39979**	2.934717
Garden	**-8.43576**	0.290577	**-7.06061**	0.118095
Football	**-27.1216**	0.654296	**-3.01393**	0.364628
MissA	**-9.00822**	**-0.1036**	**-0.51261**	**-0.18546**
Claire	**-39.902**	**-0.00963**	**-0.3236**	**-0.03849**
Sales	**-26.4823**	**-0.0348**	**-0.19016**	**-0.05219**

TABLE V: Average Speed Improvement ratio (point reduction ratio) and average MAE changed percentage $[(MAE_{KCDS} - MAE_{CDS/SCDS})/MAE_{CDS/SCDS}] \times 100\%$.

REFERENCES

[1] T. Koga, K. Iinuma, A. hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing", in Proc. Nat. Telecommun. Conf., New Orleans, L.A., Nov.-Dec. 1981, pp. G5.3.1-G5.3.5.

[2] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation", IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438-443, Aug 1994.

[3] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation", IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313-317, Jun 1996.

[4] 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.

[5] C. H. Cheung, and L. M. Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", IEEE Trans, Circuits Syst. Video Technol., vol. 12, no. 12, Dec 2002.

[6] C. H. Cheung and L. M. Po, "A novel small-cross-diamond search algorithm for fast video coding and videoconferencing applications", in Proc. IEEE ICIP, Sept. 2002.

[7] Yao Nie; Kai-Kuang Ma, "Adaptive rood pattern search for fast block-matching motion estimation", IEEE trans. Image Processing, vol. 11, pp.1442-1449, Dec 2002.