11

A Novel Prediction-Based Asymmetric Fast Search Algorithm for Video Compression

Chung-Ming Kuo[1], Nai-Chung Yang[1], I-Chang Jou[1] and Chaur-Heh Hsieh[2]
[1]Department of Information Engineering, I-Shou University Dahsu, 840, Kaohsiung,
[2]Dept. of Computer and Communication Engineering, Ming Chung University
Gui-Shan, 333, Taoyuan,
Taiwan, R.O.C.

1. Introduction

The successive frames of a video sequence are highly correlated. Therefore, by reducing temporal redundancy a high compression ratio can be achieved. Motion estimation, which reduces temporal redundancy, plays an important role in video coding systems such as H.26x and MPEG-x. The block-matching algorithm (BMA) is the most popular and widely used algorithm for motion estimation due to its simplicity and reasonable performance. In BMA, an image frame is divided into non-overlapping rectangular blocks with equal or variable block sizes. The pixels in each block are assumed to have the same motion. The motion vector (MV) of a block is estimated by searching for its best match within a search window in the previous frame. The distortion between the current block and each searching block is employed as a matching criterion. The resulting MV is used to generate a motion-compensated prediction block. The motion-compensated prediction difference blocks (called residue blocks) and the MVs are encoded and then sent to the decoder. Among the various BMAs, the full search (FS) is global optimal and most straightforward algorithm because it searches the entire search window for the best matching block. However, its only drawback is a heavy computational load.
To address this drawback, many fast BMAs have been proposed in the literature since 1981, such as the three-step search (TSS) (Koga et al., 1981), cross search (Ghanbari, 1990), new three-step search (NTSS) (Li et al., 1994), block-based gradient descent search (Liu & Feig,1996), four-step search (Po & Ma, 1996), diamond search (DS) (Zhu & Ma, 2000), hexagon-based search (HEXBS) (Zhu et al., 2002), and the cross-diamond search (CDS) (Cheung & Po, 2002), etc. The primary assumption of most fast BMAs is that the block distortion is monotonic over the search range, implying that the distortion decreases monotonically as the search position moves toward the minimum distortion point. Therefore, the best match point can be found by following the distortion trend without checking all search points in the search window. Consequently, these fast BMAs use various search patterns to reduce the number of search points, thereby speeding up the search. In addition, some fast BMAs (Liu & Zaccarin, 1993; Yu et al., 2001; Bierling, 1988) speed up the search by sub-sampling the block pixels on distortion computation and/or by sub-sampling the search points in the search window. Furthermore, since there is a high correlation between a current block and its adjacent blocks in spatial and/or temporal domains, the

current block's MV can be predicted by using the MVs of adjacent blocks. The Prediction Search Algorithm (PSA) (Luo et al., 1997) computes the mean value of MVs as the predicted motion vector (PMV) using 3 neighboring blocks (up, up right, left) in the current frame, and then starts the search algorithm from the location of the PMV. Xu et al. proposed a prediction scheme (Xu et al., 1999) in which four blocks in the previous frame are used to compute the PMV in order to enhance the traditional fast BMAs. The Adaptive Rood Pattern Search (ARPS) (Yao & Ma, 2002) uses only the previous MV on the left in the current frame as the PMV, while the search center and the pattern size are re-defined accordingly. C.-M. Kuo et al. use an adapted Kalman filter to predict the MV (Kuo et al., 2002), which greatly improves prediction accuracy while maintaining the trend of the MVs. In addition, a lot of similar models for fast BMAs have been proposed in recent years (Chimienti et al., 2002; Namuduri, 2004; Ahmad et al., 2006; Kuo et al., 2006 and 2009). In general, the common feature in all fast BMAs is the trade-off between quality and search speed. Increasing speed as much as possible, while preserving quality, is the major goal of all fast BMAs.

In this paper, a novel fast BMA is developed. In this new approach, we effectively use the information of the matching error as well as the center-biased characteristic in order to greatly minimize the search points while maintaining high quality. The experimental results show that the proposed method yields a very promising performance.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the intrinsic problems of some traditional fast BMAs. The details of the proposed DAS are given in Section 3. In Section 4, the DAS algorithms with prediction and prejudgment (DASp & DASpb, respectively) are introduced. Section 5 comprises a discussion of the experimental results. Finally, conclusions are provided in Section 6.

2. Problem formulaton

This section addresses some important issues about conventional fast BMAs. Herein we also present a framework for the proposed method.

Issue 1: The design of search pattern is usually a tradeoff between fast (large) and small motions

In some of the earlier methods (e.g. TSS and 2Dlog) that consider the possibility of fast motion, the initial search pattern is quite large and refined by gradually decreasing the search size of the search pattern. Afterwards, most fast BMAs adopt a two-stage, or coarse-to-fine, search strategy in order to accommodate various possible video types. In the coarse-to-fine search strategy, a large pattern is first used to find the possible position quickly and then switch to a small pattern to pinpoint the precise location.

We summarize the minimum search points (MSP) needed to locate an MV for some representative fast BMAs as follows:

1. TSS checks 9 points in its first step, and then 8 points in the two subsequent steps respectively. Thus MSP of the algorithm is 25 points.
2. NTSS uses nine checking points of TSS plus eight center-biased points in its first step to favor blocks with small motion. Therefore, the MSP of NTSS is 17 points.
3. DS adopts a nine-point diamond-shaped search pattern, referred to as the large diamond search pattern (LDSP), in the coarse search stage. And then a small diamond search pattern (SDSP) containing four nearest points around the center is applied in the fine stage. The MSP of DS is 13 and therefore DS outperforms NTSS in terms of search speed.

4. HEXBS adopts a 7-point large hexagonal pattern in its first step. If the minimum block distortion (MBD) occurs at the center of the hexagonal pattern, an additional 4-point small hexagonal pattern around the center is analyzed to determine the MV. The lower bound of HEXBS is 11 points and therefore HEXBS outperforms DS in terms of search speed.
5. CDS utilizes a more compact nine-point cross-shaped pattern (CSP) in its first step. If the MBD occurs at the center of the cross-shaped pattern, the search discontinues. Thus, CDS requires at least 9 search points, and therefore CDS outperforms HEXBS, on average, in terms of search speed.

However, under current two-stage search strategy, it is very difficult to further improve the search speed. In the present study, we attempt to break the two-stage strategy to further improve search efficiency.

Issue 2: The search pattern is usually symmetric, and the magnitude of block matching error is not effectively used

For most fast BMAs, the origin of search is usually set at the center of the search window and the search occurs according to a symmetric pattern. After comparison, the new center is set at the point with the least amount of block distortion, and then generates a new symmetric pattern for the next search. This procedure continues until the conditions of convergence are satisfied. However, there are two main drawbacks in such a procedure. First, in BMA, the most important assumption is the monotonic error surface. However, the design of the symmetrical pattern assumes that the direction of convergence is equally alike in each direction with respect to the search center. Therefore, the monotonic property is not properly used. If the search direction can be correctly determined, the search speed will be further improved. Second, for most BMAs, the block matching error is used to compare and find the best match. Generally, the magnitude of matching error is not effectively used. We believe that observing the magnitude variation can provide the direction of convergence and be used to enhance search speed.

Issue 3: Prediction is very useful for large motion or frame skipping

In some applications, there exists a large motion between adjacent frames due to fast motion or frame skipping. In such cases, the center-biased characteristic is not sufficiently satisfied, and therefore using a large search pattern should be more efficient. In fact, this inference is not completely true. According to our extensive experiments, whatever search pattern is used, large or small, the number of search points increases significantly. Instead of search pattern design, the prediction scheme seems to be more appropriate for reducing such bias.

From the above discussion, we can conclude that search speed is highly dependent on the design of the search pattern, and the MSP dominates search speed. To improve search speed, the MSP must be decreased. However, under the two-stage search strategy, it is very difficult to further reduce the MSP. Therefore, we have made a break with the two-stage strategy and have adopted a very compact center-biased search pattern with a directional search strategy in our study. Although using a compact center-biased search pattern decreases the MSP, it is easier to be trapped in a local minimum. Meanwhile, the number of checking points may remarkably increase for blocks with a larger MV. Using a prediction scheme is a good solution to this problem. A prediction scheme involves estimating the MV for the current block. As previously mentioned, there are many ways to predict the MV. A better prediction scheme may result in a better performance, but improvement is noticeable even with a simple one.

Based on the above discussion, utilizing a compact center-biased search pattern to favor a small MV and incorporating it with a prediction scheme to benefit a larger MV appears to be the best strategy for achieving a fast BMA. Therefore, we propose using the DAS and DASp algorithms.

3. Directional asymmetric search

The center-biased characteristic has been reported and widely used in many studies [3, 6, 8, 12-14]. It means that most MVs are very close to zero motion. Thus, the best search strategy is to search from the center of the search window and its nearest neighbors. On the other hand, although the hypothesis of monotonic error surface is not always true, it holds true primarily in the nearby region around the minimum error point (global or local). This implies that the minimum error point can be found along the direction of the block error from the highest to the lowest point. As long as the error direction is known, only the points along the search path need to be checked. Therefore, an asymmetric search pattern is more efficient in subsequent steps.
Some key issues about DAS are addressed below.

3.1 Error direction determination

Most fast BMAs find the MBD in each step using symmetric patterns. The location of the MBD in the current step is the center of the next step until the MBD occurs at the center. Under this condition, the information regarding block distortions has not been effectively used. Thus far, for most fast BMAs, it is used only for finding the MBD. Actually, it reveals not only the MBD but also the error direction.
In our study, the error direction (or search direction) in each step is defined as the direction from the location of the maximum block distortion toward the minimum block distortion. Herein S denotes a search pattern, $BD(k)$ denotes the block distortion of the search point k, and $k \in S$ is a point in S. The search direction for a specific step is defined as:

$$\vec{d} = \overrightarrow{(\arg\underset{k\in S}{Max}(BD(k)))\ \ (\arg\underset{k\in S}{Min}(BD(k)))} \tag{1}$$

As shown in Fig 1., the arrowheads designate the positions of the MBD and the arrow tails indicate the positions of the maximum block distortion. The search direction provides a very good clue regarding the approach of the final MV.

3.2 Proper search pattern selection

As shown in Fig. 1., the proposed DAS consists of 13 possible search patterns, 12 directional patterns and 1 initial pattern. In the first step, a compact 5-point cross pattern is selected, as the white circles in Fig. 1 (e), that is, S = {the white circles in Fig. 1 (e)}. If the MBD occurs at the center, the search discontinues; otherwise, subsequent steps are conducted and the search direction is determined by Eq. (1). According to the definition in Eq. (1), there are eight possible directions, but there are 12 directional patterns due to the different locations of the MBD. Once the search direction is obtained, the corresponding directional pattern will be used in the next step.
For each step that follows, three additional points (the orange circles in each directional pattern as shown in Fig. 1) are checked and the search direction is determined by Eq. (1) but with a different S, which contains 4 points, that is, S={the 3 additional points plus the MBD

point in the previous step}. Regardless of the number of checking points, the search direction is determined in the same way and both have eight possible directions.

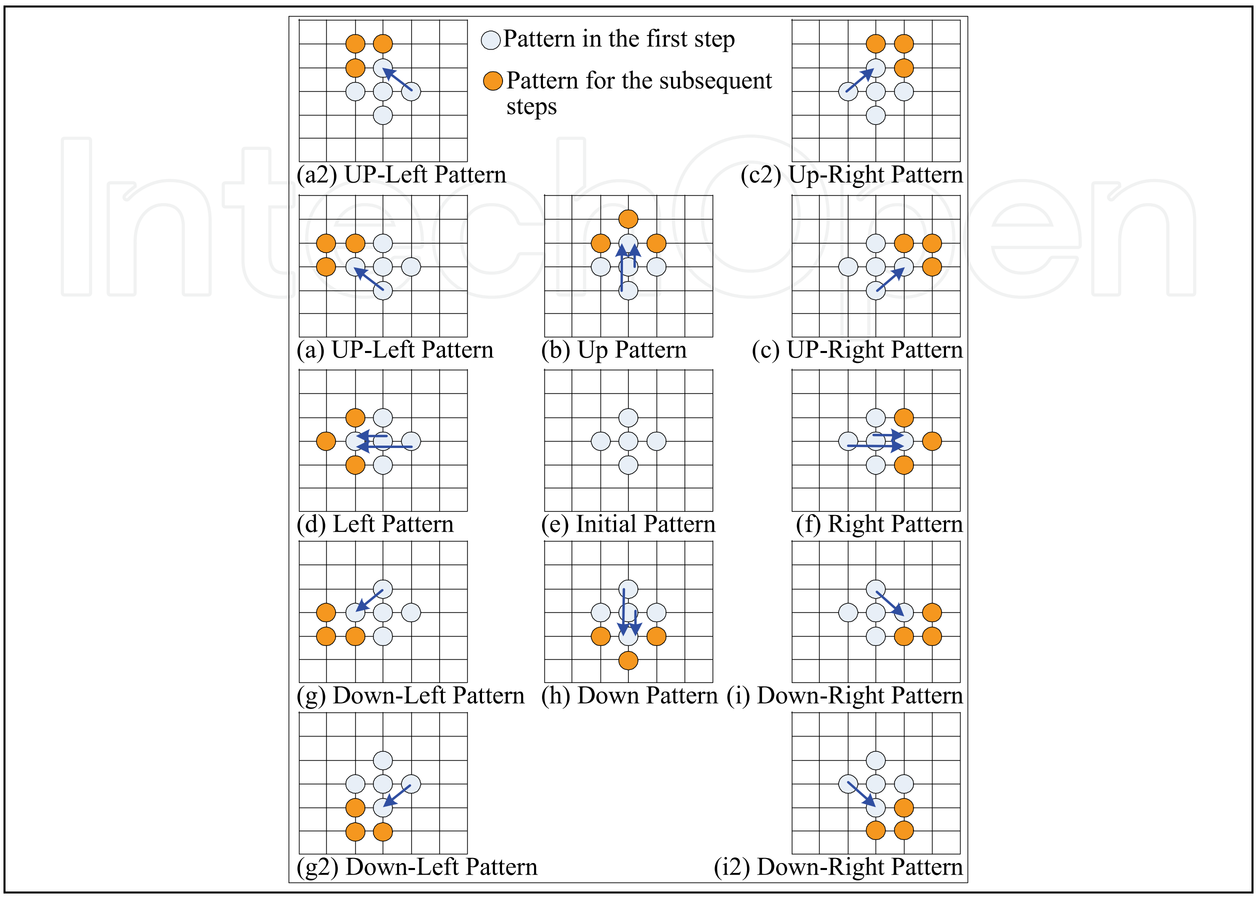


Fig. 1. Search patterns of DAS

3.3 Summary of DAS

The proposed DAS looks complicated but the underlying mechanism is quite easy. A flowchart of the DAS is shown in Fig. 3 and is summarized as follows:

Step 1. The initial cross-pattern (the white circles in Fig. 1 (e)) is centered at the origin of the search window.

Step 2. The block distortions of the 5 checking points of the cross-pattern are calculated. If the location of the MBD occurs at the center, the search discontinues and the MV is set at the center. If not, we proceed to step 3.

Step 3. Set the location of the MBD as the new center, find the error direction and select the proper search pattern for the next step accordingly.

Step 4. According to the selected pattern, three additional points (the orange circles) are checked, some of which may have already been checked. If the location of the MBD remains unchanged, the search discontinues, and the MV is set at the location of the MBD. If not, revert to step 3.

For each fast BMA, the MSP is the minimum number of checking points needed to locate the MV for a block. The lower the MSP of a fast BMA, the faster the search speed. Therefore, we have adopted a very compact 5-point cross pattern in the first step of DAS to reduce the MSP and to favor blocks with a small MV.

Fig. 2 illustrates the proposed DAS and compares it to some well-known fast BMAs. It took 16 checking points to locate the MV for DAS in this case. The required checking points for other fast BMAs are: TSS (25), NTSS (22), DS (22), CDS (24), and HEXBS (17). Obviously, DAS is the fastest algorithm in this case. Moreover, it is true in most cases and is confirmed by the experimental results of this study.

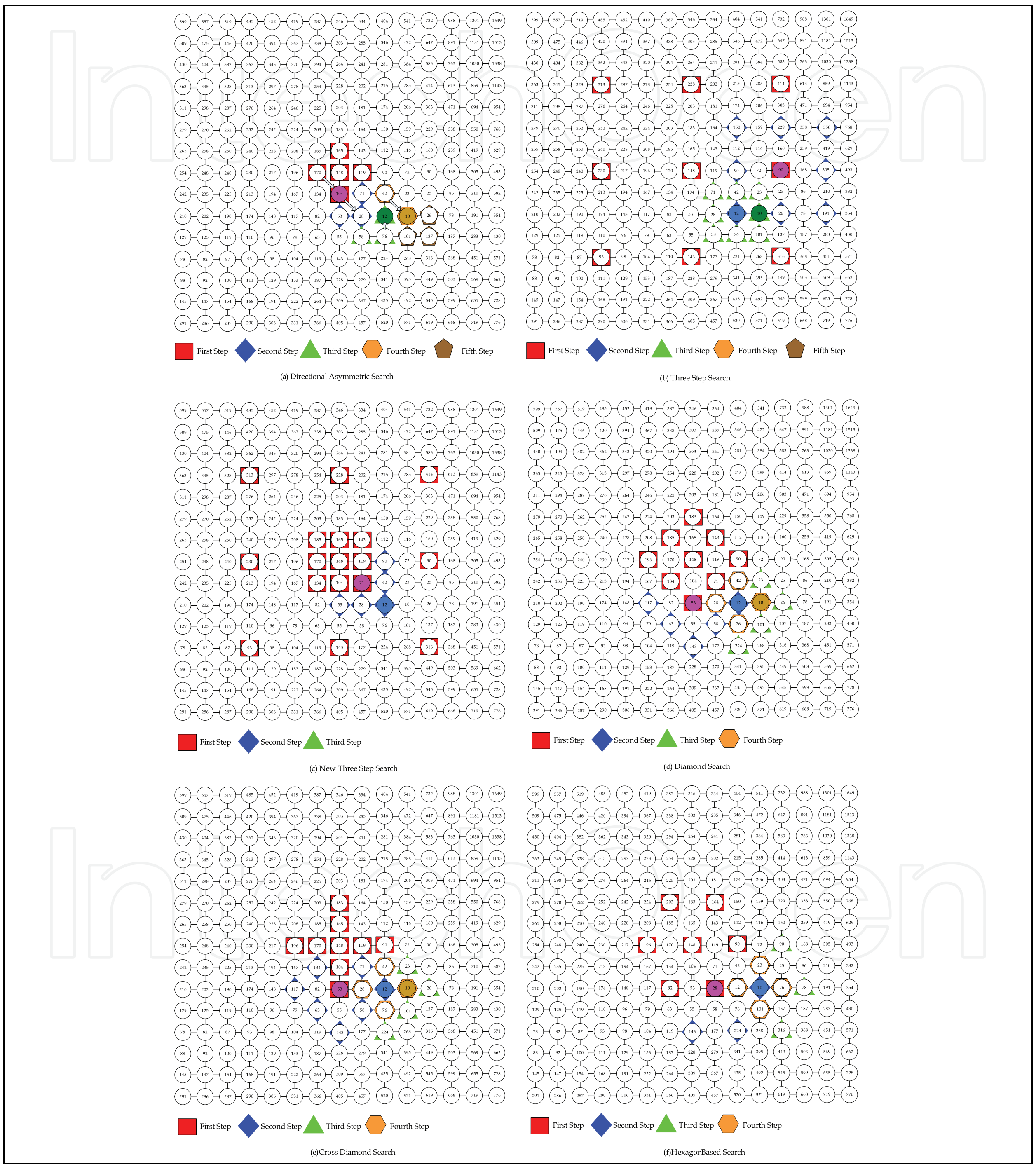


Fig. 2. A practical search example for various fast BMAs (a) DAS (b) TSS (c) NTSS (d) DS (e) CDS (f) HEXB Selected from Foreman sequence, frame 12, block 214, MV(+3,+2), block size: 16×16, search window: ±7. The number in each circle indicates the mean square error of that block.

4. Directional asymmetric search with prediction (DASP)

4.1 Prediction sheme

The DAS can be viewed as an enhanced center-biased algorithm requiring only a few checking points to locate the MV for blocks with a small MV. However, the checking points increase as the value of the MV increases. The larger the MV, the more checking points are required. Although the DAS requires a lower number of checking points compared to other fast BMAs, even in the case of a large MV, we believe this situation should be carefully addressed. A good solution for this situation is to use a prediction scheme. The initial search can start from either the PMV point or the original center point according to their block distortions.

As mentioned in Section 1, there are many ways to predict the MV for the current block [12-19]. Any kind of prediction scheme can be incorporated into the DAS. Although better prediction schemes may yield better results, we have selected the simplest one; namely, the motion vector of the previous block on the left in current frame is selected as the PMV for current block. In simulations, we will show that the prediction scheme achieves very promising performance without extra computational burdens.

The prediction scheme is incorporated into our DAS as follows and the flowchart of DASp is shown inFig. 3:

1. If the current block is a left-most block, the PMV is set to be (0, 0); otherwise the PMV is set as the MV of the previous block on the left.
2. Compute the block distortions for the position of the PMV and the center of the search window, respectively.
3. Select the position with the smaller block distortion as the search center, and start the DAS in Section 3.3 from Step 2.

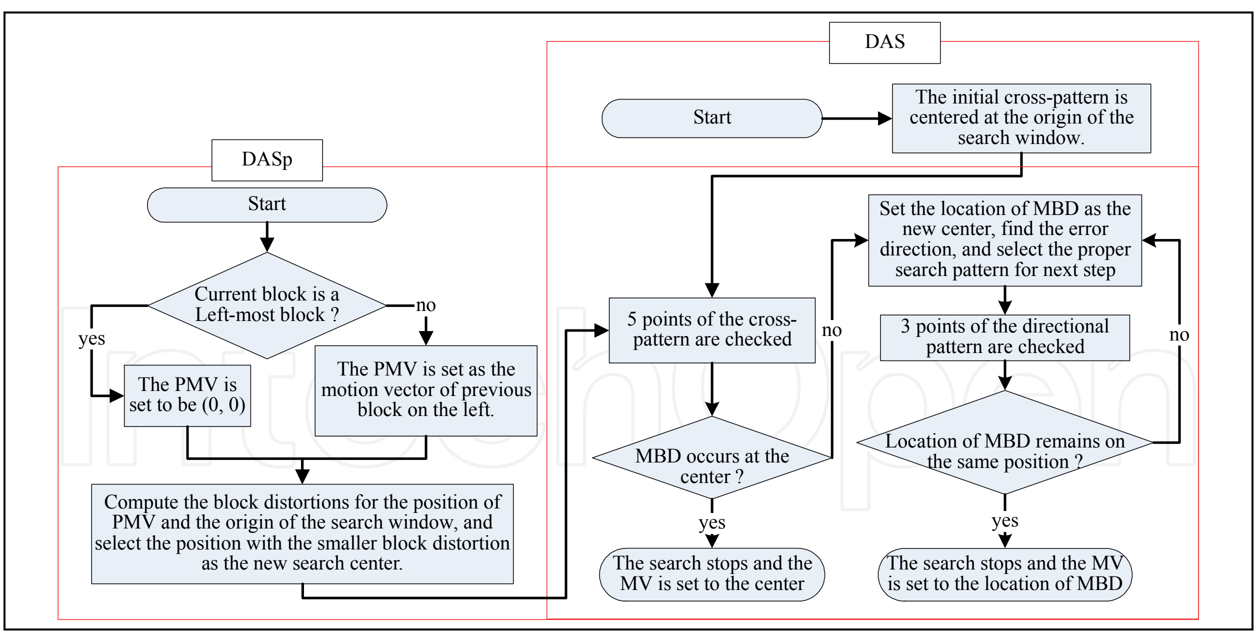


Fig. 3. Flow chart of DAS and DASp

4.2 Best-match prejudgment

According to our observations, for many stationary or quasi-stationary blocks the distortion of the best match block is close to zero. Since we found a block with a very small block distortion in our search, it was deemed unnecessary to check other points because they would not significantly improve performance, even though blocks with lower distortions may exist. We define T_{best} as the best match threshold. If the block distortion of a particular point is smaller than the T_{best} at any time during the search, the search discontinues, and that point is regarded as the best match point. Our Best-match prejudgment is different from the Zero-Motion prejudgment [14] in that the Zero-Motion prejudgment only checks the center of the search window. If the value of the block distortion is smaller than a predefined threshold, the search discontinues. Nevertheless, this situation happens not only in the zero-motion position but also in other positions, especially when a prediction scheme is incorporated. Therefore, our Best-match prejudgment reduces more checking points compared to the Zero-Motion prejudgment. With the Best-match prejudgment, the MSP of the DAS is reduced from 5 points to 1 point.

5. Experimental results and discussion

In our simulations, the FS and several famous fast BMAs are compared with the proposed DAS, DASp (DAS with prediction scheme), and DASpb (DAS with prediction scheme and the Best-match prejudgment). The fast BMAs include TSS, NTSS, DS, CDS, HEXBS, ARPS, and ARPSz (ARPS with the Zero-Motion prejudgment). To ensure a more valid comparison, these fast BMAs are classified into 3 groups as follows:

Group 1→ TSS, NTSS, DS, HEXBS, CDS, DAS

Group 2→Prediction-Based: DASp, ARPS

Group 3→Prediction-Based with prejudgment: DASpb, ARPSz

We will demonstrate the efficiency of the search pattern and search strategy either with or without prediction and prejudgment.

As shown in Table 1, nineteen popular video sequences are used in our simulations. These sequences cover a wide range of motion content and have various formats. In all simulations, the Mean Square Error (MSE) is used to measure block distortion with the block size being 16×16, and the search range ±7 pixels. The Best-match threshold T_{best} for DASpb and the Zero-motion threshold for ARPSz are all set at 1 for a fair comparison.

The simulation results are given in five aspects, which are 1) average PSNR per frame, 2) average search points per block, 3) speed-up ratio, 4) average runtime per frame, 5) runtime speed-up ratio. The average PSNR per frame and the average search points per block are summarized in Table 2, and 3, respectively. For the sake of easy comparison, Fig. 4 illustrates the contents of Table 3. In order to clearly present the differences between the

Image Sequence	Frame Size	Length		Image Sequence	Frame Size	Length
Akiyo	352 x 288	300		Foreman	352 x 288	300
Bream	352 x 288	300		Stefan	352 x 288	90
Claire	352 x 288	100		Flower_garden	352 x 240	100
Miss_America	352 x 288	100		Football	352 x 240	210
Mobile	352 x 288	300		TableTennis	352 x 240	300
Mother_daughter	352 x 288	300		Grandma	176 x 144	300
News	352 x 288	300		Silent	176 x 144	300
Paris	352 x 288	300		Suzie	176 x 144	150
Salesman	352 x 288	200		Carphone	176 x 144	300
Coastguard	352 x 288	300				

Table 1. Test sequences

simulated BMAs, the FS is not shown in Fig. 4. Table 4 shows the speed-up ratio, which is the ratio of search points per block of FS to that of other methods. In addition, the average runtime per frame is summarized in Table 5. Because the simulation is conducted on a computer under the Windows XP and the Windows XP is a multitasking operating system, under which many threads are running simultaneously sharing the CPU time, it is very hard to precisely measure the CPU time consumed by a process. Therefore, we conducted the simulation 5 times and then the mean values are shown. Fig. 6 shows the runtime speed-up ratio, which is the ratio of average runtime per frame of FS to that of other methods. The proposed methods are highlighted in Table 2, 3, 4, 5 and 6 with underlining and italics.

By observing Group 1 in Table 3, in most test sequences we can easily find that the average search points per block for each algorithm are close to the MSP of its own, especially for those sequences with small-motion content (i.e. Akiyo, Claire, News, Paris, and Grandma). The results show that the value of average search points for fast BMAs strongly depends on the selected search pattern, and using a large pattern is inefficient for sequences with small-motion content. For simplicity sake, we have used the Akiyo sequence as an example. Fig.5 gives the frame-wise comparison on the performance index of "average search points per block" for the first 300 frames and it clearly confirms our points. On the other hand, for those sequences with large-motion content, such as Coastguard, Foreman, Stefan, Flower garden, and Football, the value of average search points increases remarkably no matter which algorithms are used, except for the constant FS and TSS. However, the proposed DAS

Algorithm / Image Sequence		Group 1						Group 2		Group 3	
	FS	**TSS**	**NTSS**	**DS**	**HEXBS**	**CDS**	***DAS***	**ARPS**	***DASp***	**ARPSz**	***DASpb***
Akiyo (CIF)	42.93	42.72	42.92	42.90	42.51	42.89	**42.89**	42.88	**42.89**	42.88	**42.89**
Bream (CIF)	32.83	30.73	31.99	31.96	31.19	31.88	**31.45**	32.14	**32.35**	32.20	**32.31**
Claire (CIF)	41.32	41.22	41.30	41.30	40.99	41.26	**41.26**	41.23	**41.27**	41.27	**41.27**
Miss_America (CIF)	39.16	38.68	39.09	38.83	38.35	39.05	**39.04**	38.96	**38.88**	38.96	**38.88**
Mobile (CIF)	25.16	24.86	25.13	25.08	24.82	25.10	**25.09**	24.98	**25.09**	24.99	**25.09**
Mother_daughter (CIF)	40.34	40.12	40.26	40.23	40.04	40.20	**40.13**	39.99	**40.22**	40.07	**40.22**
News (CIF)	37.06	36.81	36.90	36.87	36.70	36.85	**36.73**	36.68	**36.77**	36.68	**36.77**
Paris (CIF)	32.13	31.91	32.05	32.02	31.86	31.99	**31.92**	31.52	**31.95**	31.52	**31.95**
Salesman (CIF)	35.70	35.53	35.67	35.61	35.53	35.66	**35.66**	35.61	**35.63**	35.61	**35.63**
Coastguard (CIF)	*30.80*	*30.46*	*30.74*	*30.72*	*30.66*	*30.72*	***30.65***	*30.74*	***30.76***	*30.74*	***30.76***
Foreman (CIF)	*31.48*	*30.81*	*31.16*	*31.02*	*30.46*	*31.00*	***30.78***	*31.01*	***31.10***	*31.01*	***31.10***
Stefan (CIF)	*24.84*	*24.30*	*24.29*	*23.57*	*23.43*	*23.48*	***22.99***	*24.35*	***24.38***	*24.35*	***24.38***
Flower_garden (SIF)	*25.38*	*24.06*	*25.13*	*25.09*	*24.44*	*25.12*	***24.79***	*24.19*	***25.06***	*24.19*	***25.06***
Football (SIF)	*25.22*	*24.68*	*24.89*	*24.72*	*24.52*	*24.69*	***24.42***	*24.69*	***24.66***	*24.69*	***24.66***
TableTennis (SIF)	28.32	25.26	27.59	27.94	27.39	27.91	**27.35**	27.06	**27.68**	27.06	**27.68**
Grandma (QCIF)	42.35	42.34	42.34	42.34	42.32	42.34	**42.34**	42.25	**42.34**	42.28	**42.34**
Silent (QCIF)	35.64	35.57	35.58	35.56	35.42	35.53	**35.45**	35.39	**35.48**	35.41	**35.48**
Suzie (QCIF)	36.58	36.38	36.55	36.50	36.16	36.43	**36.41**	36.43	**36.42**	36.43	**36.42**
Carphone (QCIF)	32.51	32.17	32.30	32.26	31.94	32.19	**32.13**	31.93	**32.14**	31.93	**32.14**
Total Average	33.67	33.08	33.47	33.40	33.09	33.38	**33.24**	33.26	**33.42**	33.28	**33.42**

Table 2. Average PSNR per frame (dB)

is still the fastest. This indicates that the two-stage (coarse-to-fine) search strategy requires more search points due to its larger MSP. The Coastguard sequence is another example. Fig.6 gives the frame-wise comparison on the performance index of "average search points per block" for the first 300 frames. It clearly shows that the proposed DAS requires the least amount of search points among all the BMAs in Group 1 for almost all frames. An exception is frames 69 to 73, where HEXBS is the fastest one. Nevertheless, by observing Group 1 in Table 2, we found that the PSNR of the proposed DAS reveals a little degradation for the large-motion sequences mentioned above. Moreover, Fig. 7 shows the average magnitude of MV per block calculated by FS. It can be seen that the average magnitude values of frames 69 to 73 are much larger than on other frames. Meanwhile, the PSNRs of the proposed DAS are worst in these frames. Compared with the motion estimation algorithms with large search pattern, the results strongly indicate that using only small patterns can easily be trapped into local minimum especially for large-motion sequences, and therefore we should carefully address this disadvantage.

Algorithm (MSP) / Image Sequence			Group 1						Group 2		Group 3	
		FS (255)	TSS (25)	NTSS (17)	DS (13)	HEXBS (11)	CDS (9)	*DAS* (5)	ARPS (5)	*DASp* (5)	ARPSz (1)	*DASpb* (1)
Akiyo	(CIF)	225	25.00	17.13	13.08	11.04	9.12	**5.12**	5.35	**5.07**	2.70	**2.60**
Bream	(CIF)	225	25.00	21.29	16.80	13.47	13.15	**8.91**	8.91	**6.16**	5.84	**3.56**
Claire	(CIF)	225	25.00	17.54	13.30	11.16	9.47	**5.35**	6.22	**5.26**	4.53	**4.20**
Miss_America	(CIF)	225	25.00	21.48	17.53	12.81	12.44	**8.01**	8.78	**6.73**	8.97	**6.73**
Mobile	(CIF)	225	25.00	19.65	14.30	11.56	11.01	**7.56**	7.94	**5.54**	7.95	**5.52**
Mother_daughter	(CIF)	225	25.00	19.62	14.59	11.83	11.36	**6.64**	7.60	**6.28**	7.06	**5.99**
News	(CIF)	225	25.00	17.80	13.67	11.38	10.07	**5.79**	6.22	**5.58**	4.29	**3.77**
Paris	(CIF)	225	25.00	17.63	13.47	11.26	9.72	**5.58**	6.06	**5.43**	5.18	**4.62**
Salesman	(CIF)	225	25.00	18.09	13.60	11.29	9.87	**6.07**	6.56	**5.48**	6.58	**5.48**
Coastguard	(CIF)	*225*	*25.00*	*21.24*	*17.52*	*13.68*	*16.62*	***10.04***	*8.64*	***6.45***	*9.24*	***6.45***
Foreman	(CIF)	*225*	*25.00*	*23.04*	*18.40*	*13.78*	*16.79*	***10.86***	*9.31*	***7.56***	*9.73*	***7.53***
Stefan	(CIF)	*225*	*25.00*	*24.18*	*17.99*	*13.96*	*16.75*	***10.70***	*8.66*	***6.95***	*9.20*	***6.95***
Flower_garden	(SIF)	*225*	*25.00*	*23.42*	*18.26*	*14.20*	*16.47*	***10.53***	*9.35*	***6.75***	*9.77*	***6.68***
Football	(SIF)	*225*	*25.00*	*25.77*	*20.87*	*15.24*	*20.30*	***13.32***	*10.86*	***9.92***	*11.46*	***9.92***
TableTennis	(SIF)	225	25.00	20.53	16.53	13.11	14.18	**8.50**	8.18	**6.47**	8.54	**6.47**
Grandma	(QCIF)	225	25.00	17.73	13.46	11.25	9.73	**5.40**	6.27	**5.40**	6.02	**5.19**
Silent	(QCIF)	225	25.00	17.86	13.65	11.35	10.04	**5.79**	6.65	**5.68**	4.08	**3.49**
Suzie	(QCIF)	225	25.00	18.76	14.23	11.61	10.87	**6.55**	7.19	**6.02**	5.83	**4.96**
Carphone	(QCIF)	225	25.00	19.71	15.07	12.03	12.06	**7.41**	8.52	**6.84**	8.08	**6.41**
Total Average		225	25.00	20.13	15.60	12.42	12.63	**7.80**	7.75	**6.29**	7.11	**5.61**

Table 3. Average search points per block

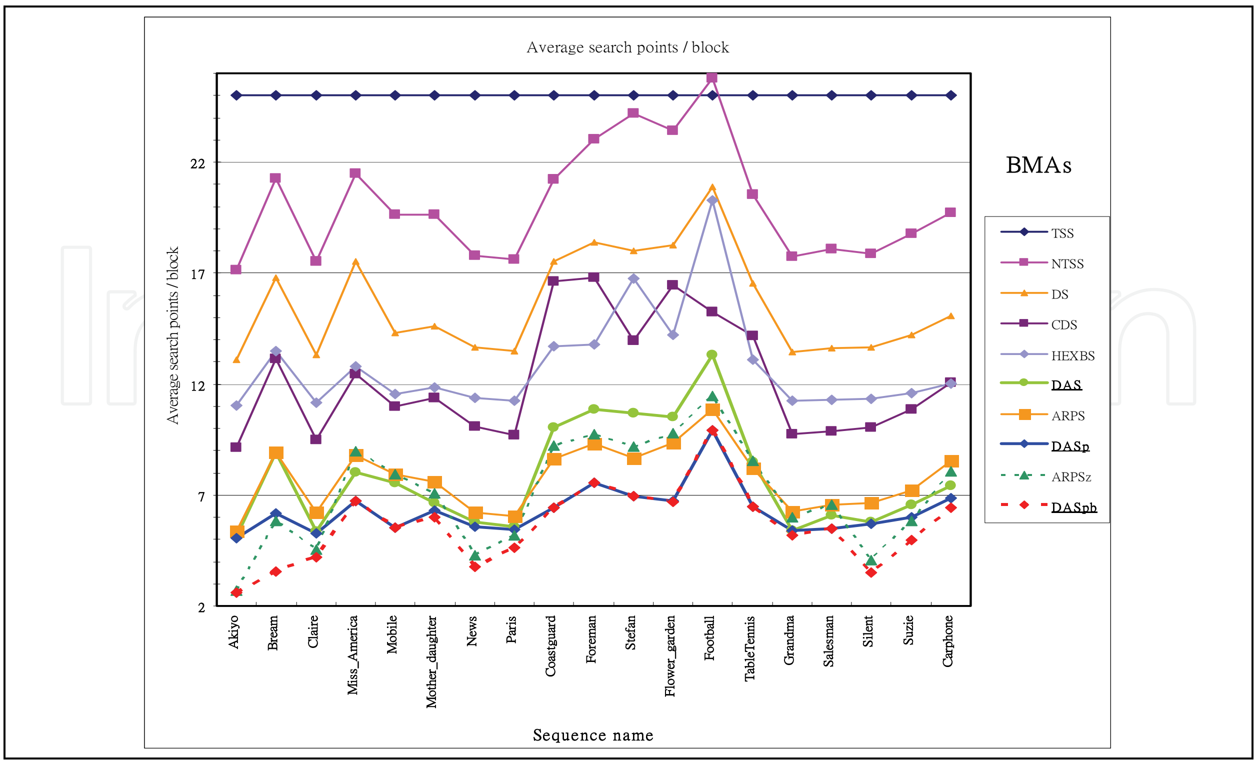


Fig. 4. Average search points per block for all test sequences

Algorithm / Image Sequence		Group 1						Group 2		Group 3	
	FS	**TSS**	**NTSS**	**DS**	**HEXBS**	**CDS**	***DAS***	**ARPS**	***DASp***	**ARPSz**	***DASpb***
Akiyo (CIF)	1.00	9.00	13.13	17.20	20.38	24.67	**43.95**	42.06	**44.38**	83.33	**86.54**
Bream (CIF)	1.00	9.00	10.57	13.39	16.70	17.11	**25.25**	25.25	**36.53**	38.53	**63.20**
Claire (CIF)	1.00	9.00	12.83	16.92	20.16	23.76	**42.06**	36.17	**42.78**	49.67	**53.57**
Miss_America (CIF)	1.00	9.00	10.47	12.84	17.56	18.09	**28.09**	25.63	**33.43**	25.08	**33.43**
Mobile (CIF)	1.00	9.00	11.45	15.73	19.46	20.44	**29.76**	28.34	**40.61**	28.30	**40.76**
Mother_daughter (CIF)	1.00	9.00	11.47	15.42	19.02	19.81	**33.89**	29.61	**35.83**	31.87	**37.56**
News (CIF)	1.00	9.00	12.64	16.46	19.77	22.34	**38.86**	36.17	**40.32**	52.45	**59.68**
Paris (CIF)	1.00	9.00	12.76	16.70	19.98	23.15	**40.32**	37.13	**41.44**	43.44	**48.70**
Salesman (CIF)	1.00	9.00	12.44	16.54	19.93	22.80	**37.07**	34.30	**41.06**	34.19	**41.06**
Coastguard (CIF)	*1.00*	*9.00*	*10.59*	*12.84*	*16.45*	*13.54*	***22.41***	*26.04*	***34.88***	*24.35*	***34.88***
Foreman (CIF)	*1.00*	*9.00*	*9.77*	*12.23*	*16.33*	*13.40*	***20.72***	*24.17*	***29.76***	*23.12*	***29.88***
Stefan (CIF)	*1.00*	*9.00*	*9.31*	*12.51*	*16.12*	*13.43*	***21.03***	*25.98*	***32.37***	*24.46*	***32.37***
Flower_garden (SIF)	*1.00*	*9.00*	*9.61*	*12.32*	*15.85*	*13.66*	***21.37***	*24.06*	***33.33***	*23.03*	***33.68***
Football (SIF)	*1.00*	*9.00*	*8.73*	*10.78*	*14.76*	*11.08*	***16.89***	*20.72*	***22.68***	*19.63*	***22.68***
TableTennis (SIF)	1.00	9.00	10.96	13.61	17.16	15.87	**26.47**	27.51	**34.78**	26.35	**34.78**
Grandma (QCIF)	1.00	9.00	12.69	16.72	20.00	23.12	**41.67**	35.89	**41.67**	37.38	**43.35**
Silent (QCIF)	1.00	9.00	12.60	16.48	19.82	22.41	**38.86**	33.83	**39.61**	55.15	**64.47**
Suzie (QCIF)	1.00	9.00	11.99	15.81	19.38	20.70	**34.35**	31.29	**37.38**	38.59	**45.36**
Carphone (QCIF)	1.00	9.00	11.42	14.93	18.70	18.66	**30.36**	26.41	**32.89**	27.85	**35.10**
Total Average	1.00	9.00	11.34	14.71	18.29	18.84	**31.23**	30.03	**36.62**	36.15	**44.27**

Table 4. Speed-up ratio with respect to FS

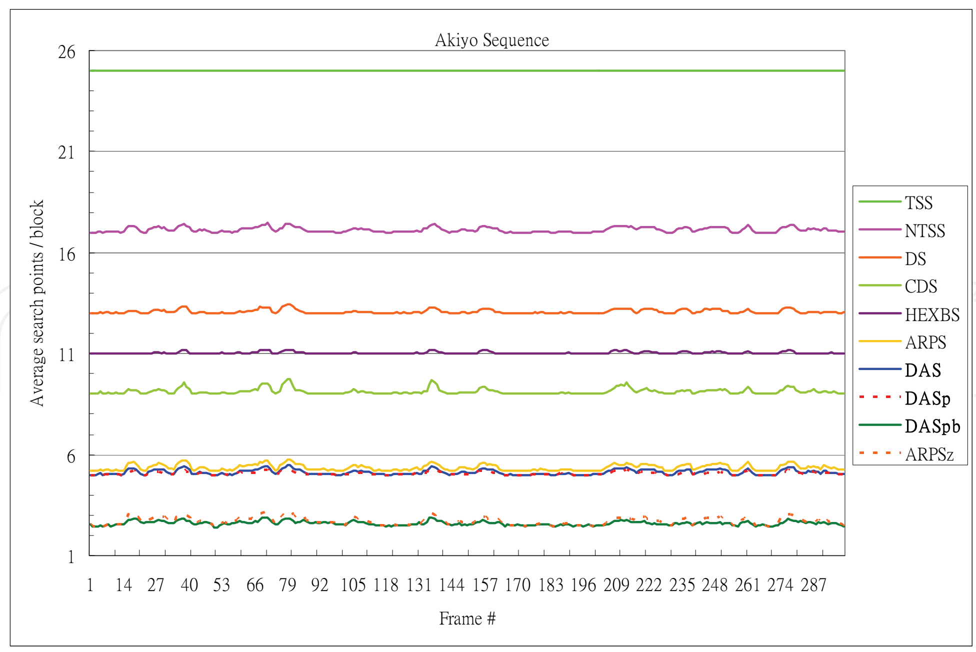


Fig. 5. Frame-wise average search points comparison between various BMAs on CIF sequence "Akiyo"

Algorithm / Image Sequence			Group 1						Group 2		Group 3	
		FS	**TSS)**	**NTSS**	**DS**	**HEXBS**	**CDS**	***DAS***	**ARPS**	***DASp***	**ARPSz**	***DASpb***
Akiyo	(CIF)	295.98	33.58	23.71	18.52	15.79	13.26	**7.56**	7.92	**7.62**	4.39	**4.26**
Bream	(CIF)	295.85	34.29	29.14	23.77	19.12	18.49	**12.87**	13.35	**8.88**	8.84	**5.52**
Claire	(CIF)	297.08	34.06	24.24	18.47	15.62	13.51	**7.63**	9.14	**7.81**	6.79	**6.31**
Miss_America	(CIF)	295.71	34.05	29.30	24.14	17.84	17.12	**12.41**	12.58	**10.00**	13.06	**9.46**
Mobile	(CIF)	296.07	34.18	27.02	20.26	16.61	15.58	**11.18**	12.07	**8.13**	12.11	**8.22**
Mother_daughter	(CIF)	295.47	34.19	27.23	20.84	16.98	16.08	**9.78**	11.28	**9.17**	10.56	**8.68**
News	(CIF)	295.32	34.34	24.50	19.30	16.22	14.32	**8.54**	9.12	**8.15**	6.51	**5.83**
Paris	(CIF)	295.59	34.06	24.24	19.01	15.87	13.91	**8.41**	9.01	**8.15**	7.76	**7.04**
Salesman	(CIF)	295.86	34.13	24.78	18.92	16.03	14.51	**9.33**	9.60	**8.20**	9.85	**7.82**
Coastguard	(CIF)	*296.43*	*34.46*	*29.05*	*24.41*	*19.36*	*22.98*	***14.52***	*12.10*	***9.65***	*13.24*	***9.79***
Foreman	(CIF)	*296.24*	*34.26*	*31.74*	*25.79*	*19.65*	*23.81*	***15.74***	*13.39*	***11.07***	*14.03*	***11.19***
Stefan	(CIF)	*297.65*	*34.15*	*32.98*	*25.12*	*19.87*	*23.93*	***14.90***	*12.19*	***10.47***	*13.39*	***10.38***
Flower_garden	(SIF)	*246.64*	*28.37*	*26.24*	*21.08*	*16.32*	*19.26*	***12.52***	*10.93*	***8.42***	*11.39*	***8.08***
Football	(SIF)	*246.52*	*28.57*	*29.71*	*23.89*	*17.99*	*23.60*	***15.93***	*13.07*	***12.33***	*13.78*	***11.91***
TableTennis	(SIF)	246.84	28.88	23.63	19.23	15.68	16.78	**10.45**	9.95	**8.14**	10.21	**8.46**
Grandma	(QCIF)	74.99	8.31	10.91	5.98	5.11	3.90	**1.54**	2.09	**1.75**	1.85	**1.80**
Silent	(QCIF)	75.49	8.97	5.59	4.19	2.52	2.74	**2.54**	3.29	**2.16**	2.10	**1.92**
Suzie	(QCIF)	75.18	8.87	5.89	3.57	3.48	3.60	**1.98**	1.65	**1.84**	2.54	**1.31**
Carphone	(QCIF)	75.29	8.43	7.28	8.12	6.76	6.52	**2.62**	3.90	**2.41**	3.41	**2.41**
Note: the frame size is different for different formats												

Table 5. Average runtime per frame (ms)

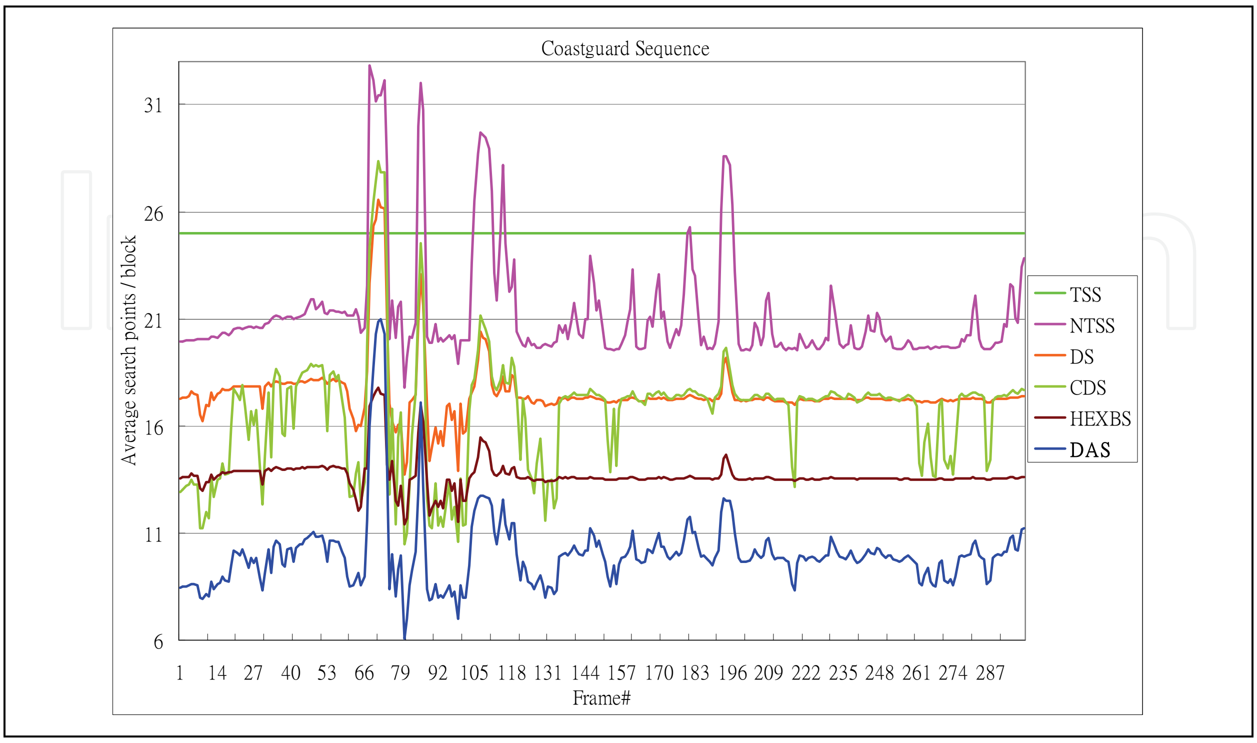


Fig. 6. Frame-wise average search points comparison between various BMAs on CIF sequence "Coastguard"

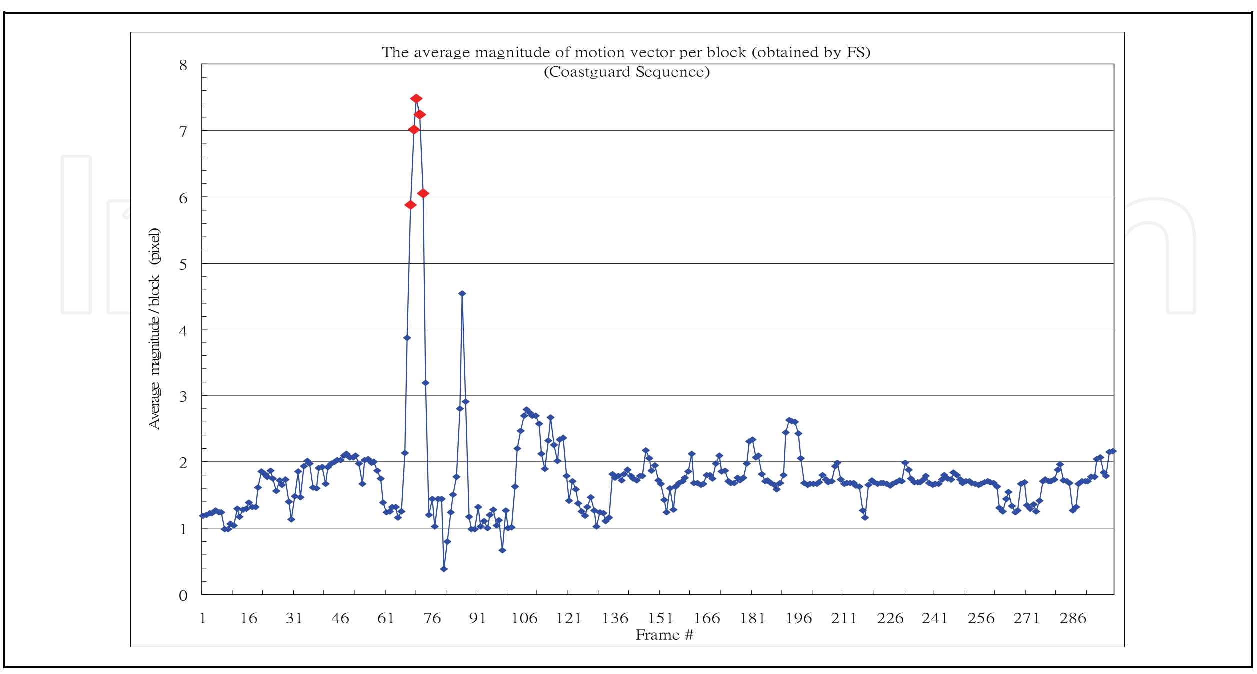


Fig. 7. The average magnitude of motion vector per block for Coastguard Sequence

Algorithm / Image Sequence		FS	Group 1						Group 2		Group 3	
		FS	TSS	NTSS	DS	HEXBS	CDS	*DAS*	ARPS	*DASp*	ARPSz	*DASpb*
Akiyo	(CIF)	1.00	8.81	12.48	15.99	18.74	22.32	**39.15**	37.37	**38.86**	67.47	**69.45**
Bream	(CIF)	1.00	8.63	10.15	12.45	15.47	16.00	**22.99**	22.16	**33.33**	33.48	**53.61**
Claire	(CIF)	1.00	8.72	12.26	16.08	19.02	21.99	**38.94**	32.49	**38.02**	43.73	**47.10**
Miss_America	(CIF)	1.00	8.68	10.09	12.25	16.58	17.27	**23.84**	23.50	**29.58**	22.64	**31.26**
Mobile	(CIF)	1.00	8.66	10.96	14.61	17.83	19.01	**26.49**	24.54	**36.42**	24.46	**36.04**
Mother_daughter	(CIF)	1.00	8.64	10.85	14.18	17.40	18.38	**30.20**	26.20	**32.22**	27.97	**34.03**
News	(CIF)	1.00	8.60	12.06	15.30	18.21	20.62	**34.56**	32.38	**36.25**	45.35	**50.65**
Paris	(CIF)	1.00	8.68	12.20	15.55	18.62	21.24	**35.13**	32.82	**36.25**	38.10	**41.98**
Salesman	(CIF)	1.00	8.67	11.94	15.64	18.46	20.39	**31.71**	30.83	**36.08**	30.04	**37.86**
Coastguard	(CIF)	*1.00*	*8.60*	*10.21*	*12.14*	*15.31*	*12.90*	***20.41***	*24.50*	***30.72***	*22.39*	***30.29***
Foreman	(CIF)	*1.00*	*8.65*	*9.33*	*11.49*	*15.08*	*12.44*	***18.82***	*22.12*	***26.77***	*21.12*	***26.47***
Stefan	(CIF)	*1.00*	*8.72*	*9.03*	*11.85*	*14.98*	*12.44*	***19.98***	*24.41*	***28.42***	*22.23*	***28.66***
Flower_garden	(SIF)	*1.00*	*8.69*	*9.40*	*11.70*	*15.12*	*12.81*	***19.70***	*22.57*	***29.30***	*21.66*	***30.53***
Football	(SIF)	*1.00*	*8.63*	*8.30*	*10.32*	*13.70*	*10.45*	***15.47***	*18.86*	***19.99***	*17.89*	***20.70***
TableTennis	(SIF)	1.00	8.55	10.45	12.83	15.74	14.71	**23.63**	24.81	**30.31**	24.17	**29.19**
Grandma	(QCIF)	1.00	9.02	6.87	12.53	14.67	19.23	**48.56**	35.85	**42.95**	40.51	**41.75**
Silent	(QCIF)	1.00	8.41	13.50	18.00	29.91	27.53	**29.67**	22.96	**34.98**	35.87	**39.25**
Suzie	(QCIF)	1.00	8.47	12.76	21.07	21.61	20.90	**37.87**	45.53	**40.95**	29.55	**57.52**
Carphone	(QCIF)	1.00	8.94	10.35	9.27	11.15	11.54	**28.77**	19.29	**31.30**	22.06	**31.21**
Total Average		1.00	8.67	10.69	13.86	17.24	17.48	28.73	27.54	33.30	31.09	38.82

Table 6. Runtime speed-up ratio with respect to FS

To compensate the bias due to fast motion, using a prediction scheme is a useful solution. This viewpoint is confirmed by the simulation results of DASp. With a prediction scheme, the DASp performs quite well in both search speed and PSNR. Fig. 8 gives a closer view of the Coastguard sequence and indicates that the prediction scheme greatly reduces the search points needed for each frame, especially for frames 69 to 73. In addition, Fig. 9 shows PSNR performance for frames 69 to 73. As expected, the prediction scheme effectively compensates the bias and thus greatly improves PSNR performance. It is worth mentioning that the prediction scheme does not need to be very accurate. In our study, we use only the previous block to predict the MV in the current block. However, it effectively improved PSNR performance. Furthermore, the Best-match prejudgment is quite useful for sequences with a stationary background. The Best-match threshold T_{best} is used to control the results of the Best-match prejudgment. With a higher T_{best}, the Best-match prejudgment eliminates the need for more search points, but may degrade the PSNR. In our experiments, we set a very low value (T_{best}=1) for the Best-match threshold in order to preserve high estimation accuracy (high PSNR). What is the best value of T_{best} for the tradeoff between search speed and PSNR performance is beyond the scope of this paper. Nevertheless, even with such a low value, the speed-up ratio is still significant. As shown in Table 4, the DASpb almost doubles the search speed of the DASp for Akiyo without degrading PSNR.

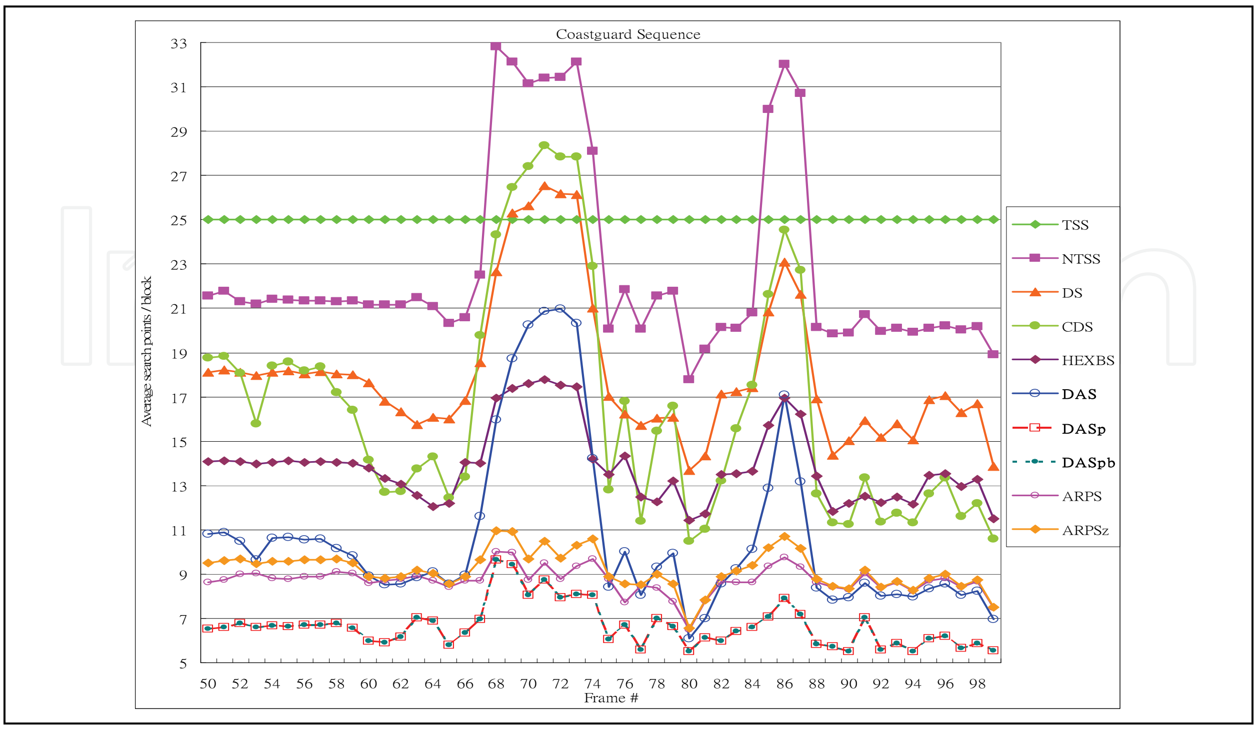


Fig. 8. Average search points / block for frame #50 to #100 of Coastguard sequence

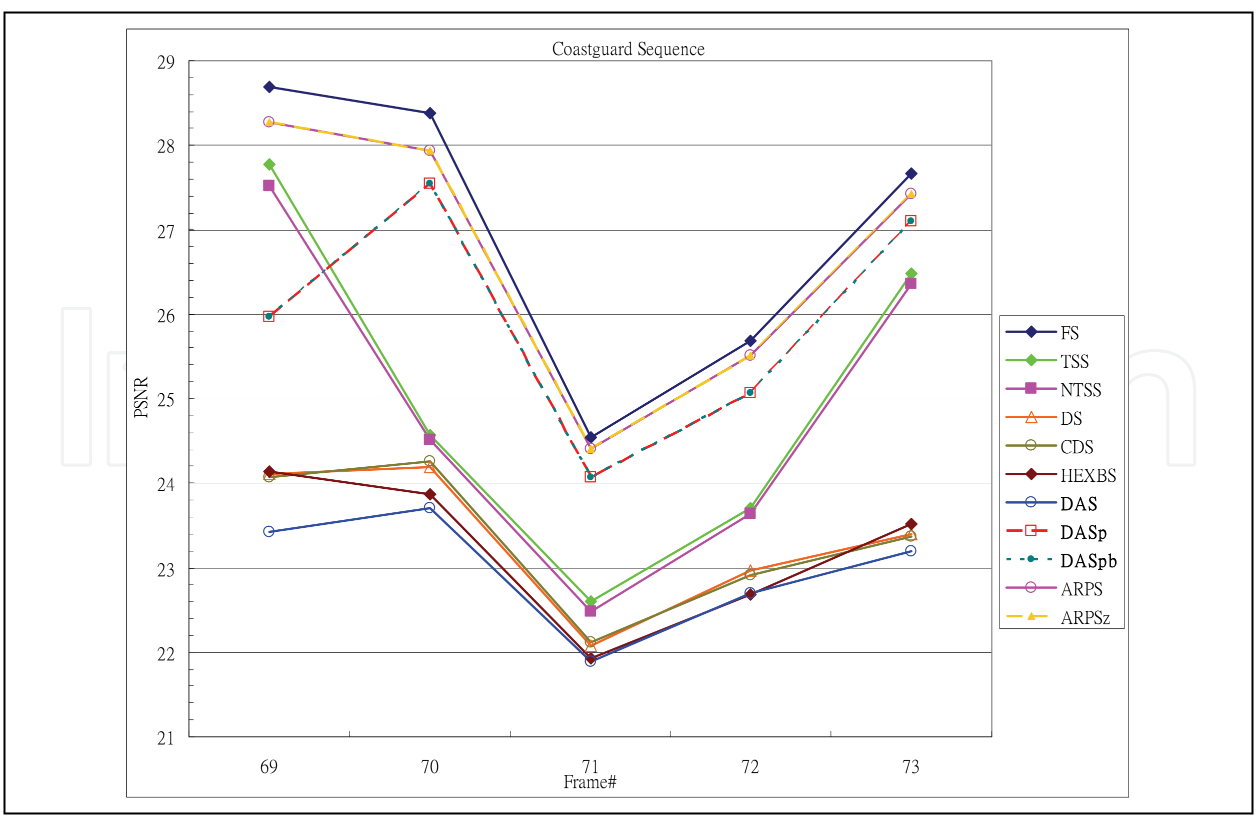


Fig. 9. The PSNR for frame #69 to #73 of Coastguard sequence

From the viewpoint of PSNR performance, Table 2 shows that the PSNR of the proposed DAS is approximately close to the average PSNR in Group 1. Furthermore, for both the DASp and DASpb, the PSNR values significantly improve, especially in fast motion sequences. In other words, for most sequences, we should enhance the property of center-bias even for those with fast motion. However, to avoid being trapped in a local minimum due to a small search pattern, the prediction of search center can effectively address this problem. For the sequences of Coastguard, Foreman, Stefan, Flower garden, and Football, Table 3 shows that the DASp improves the PSNR performance significantly. In Table 4 it can be seen that the search points for these sequences are almost the same as those sequences with a small motion. By using Best-match prejudgment, we can filter out the stationary background, thus improving search speed. Since the threshold is very low, performance degradation is minimal.

Finally, in terms of search speed, we can see from Table 3 that our methods outperform all other methods for all test sequences in the same group and the proposed DASpb has the best search speed among all BMAs with negligible degradation in the PSNR. Furthermore, the actual runtime in Table 5. confirmed the theoretical search speed in Table 3 and the runtime speed-up ratio in Table 6 roughly corresponds to the theoretical speed-up ratio in Table 4.

6. Conclusion

In this paper, we have proposed a novel fast block-matching algorithm for motion estimation. We explored the relationship between block distortions and search patterns and came up with a rule for determining search direction. With a known search direction, asymmetric search patterns are developed, and the search points on the outside of the direction were disregarded. Since the unnecessary search points are eliminated, the search speed is greatly improved.

In our study, we adopted a very compact center-biased initial pattern to minimize the required MSP. Nevertheless, it is easier to become trapped in a local minimum. We introduced a prediction scheme to address this disadvantage. The prediction scheme effectively and efficiently improves both the PSNR and search speed, especially for those sequences with fast motion (e.g. Coastguard, Foreman, Stefan, Flower garden and Football). In addition, the Best-match prejudgment was also incorporated to profit stationary and quasi-stationary blocks. The experimental results have verified our points and demonstrated the superiority of our methods. The authors would like to express their sincere thanks to the anonymous reviewers for their invaluable comments and suggestions.

7. Acknowledgement

This work was supported by the National Science Counsel of Republic of China Granted NSC 99-2221-E-214 -055.

8. References

Koga, T.; Iinuma, K.; Hirano, A.; Iijima, Y. & lshiguro, T. (1981). Motion-compensated interframe coding for video conferencing, *in Proc. Nat. Telecommunication Conf. , New Orleans*, LA, pp. G5.3.1-G5.3.5.

Ghanbari, M. (1990). The cross-search algorithm for motion estimation, *IEEE Trans. Commun.*, Vol. 38, No. 7, pp. 950-953, ISSN : 0090-6778.

Li, R.; Zeng,B. & Liou, M. L. (1994). A new three-step search algorithm for block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.* , Vol. 4, No. 4, pp. 438-442, ISSN: 1051-8215.

Liu, L. K. & Feig, E. (1996). 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. 419-422, ISSN : 1051-8215.

Po, L. M. & Ma, W. C. (1996). A novel four-step search algorithm for fast block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.* , Vol. 6, No. 3, pp. 313-317, ISSN: 1051-8215.

Zhu, S. & Ma, K. K. (2000). A new diamond search algorithm for fast block-matching motion estimation, *IEEE Trans. Image Process.* , Vol. 9, No. 2, pp. 287-290, ISSN: 1057-7149.

Zhu, C.; Lin, X. & Chau, L. P. (2002). Hexagon-based search pattern for fast block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.* , Vol. 12, No. 5, pp. 349-355, ISSN : 1051-8215.

Cheung, C. H. & Po, L. M. (2002). A novel cross-diamond search algorithm for fast block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.* , Vol. 12, No. 12, pp. 1168-1177, ISSN : 1051-8215.

Liu, B. & Zaccarin, A. (1993). New fast algorithms for the estimation of block motion vectors, *IEEE Trans. Circuits Syst. Video Technol.* , Vol. 3, No. 2, pp. 148-157, ISSN : 1051-8215.

Yu, Y.; Zhou, J. & Chen, C. W. (2001). A novel fast block motion estimation algorithm based on combined subsamplings on pixels and search candidates, *J. Vis. Commun. Image Represent.* , Vol. 12, No. 1, pp. 96-105, ISSN: 1047-3203.

Bierling, M. (1988). Displacement estimation by hierarchical block matching, *in Proc. SPIE Conf. On Vis. Commun. And Image Proc*, ISSN: 1018-8770.

Luo, L.; Zou, C.; Gao, X. & He, Z. (1997). A new prediction search algorithm for block motion estimation in video coding, *IEEE Trans. Consum. Electron.*, Vol. 43, No.1, pp. 56-61, ISSN : 0098-3063.

Xu, J. B.; Po, L. M. & Cheung, C. K. (1999). Adaptive motion tracking block matching algorithms for video coding, *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 9, No.7, ISSN : 1051-8215.

Yao, N. & Ma, K. K. (2002). Adaptive rood pattern search for fast block-matching motion estimation, *IEEE Trans. Image Process.* , Vol. 11, No. 12, pp. 1442-1449, ISSN : 1057-7149.

Kuo, C. M.; Chao, C. P. & Hsieh, C. H. (2002) A new motion estimation algorithm for video coding using adaptive kalman filter, *Real-Time Imaging,* Vol. 8, No. 5, pp. 387-398, ISSN 1077-2014.

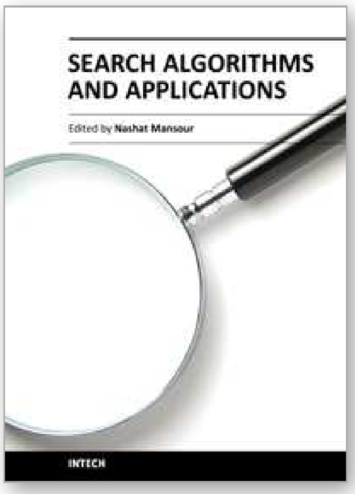
Chimienti, A.; Ferraris, C. & Pau, D. (2002). A complexity-bounded motion estimation algorithm, *IEEE Trans. Image Process.* , Vol. 11, No. 4, pp. 387-392, ISSN : 1057-7149.

Namuduri, K. R. (2004). Motion estimation using spatio-temporal contextual information, *IEEE Trans. Circuits Syst. Video Technol.* , Vol. 14, No. 8, pp. 1111-1115, ISSN : 1051-8215.

Ahmad, I.; Zheng, W.; Luo, J. & Liou, M. (2006). A fast adaptive motion estimation algorithm, *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 16, No. 3, pp. 420-438, ISSN: 1051-8215.

Kuo, C. M.; Chung, S. C. & Shih, P. Y. (2006). Kalman filtering based rate-constrained motion estimation for very low bit rate video coding, *IEEE Trans. Circuits Syst. Video Technol*., Vol. 16, No. 1, pp. 3-18, ISSN : 1051-8215.

Kuo, C. M.; Kuan, Y. H.; Hsieh, C. H. & Lee Y. H. (2009). A novel prediction-based directional asymmetric search algorithm for fast block-matching motion estimation, *IEEE Trans. Circuits Syst. Video Technol*., Vol. 19, No. 6, pp. 893-899, ISSN : 1051-8215.



Search Algorithms and Applications
Edited by Prof. Nashat Mansour

ISBN 978-953-307-156-5
Hard cover, 494 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Search algorithms aim to find solutions or objects with specified properties and constraints in a large solution search space or among a collection of objects. A solution can be a set of value assignments to variables that will satisfy the constraints or a sub-structure of a given discrete structure. In addition, there are search algorithms, mostly probabilistic, that are designed for the prospective quantum computer. This book demonstrates the wide applicability of search algorithms for the purpose of developing useful and practical solutions to problems that arise in a variety of problem domains. Although it is targeted to a wide group of readers: researchers, graduate students, and practitioners, it does not offer an exhaustive coverage of search algorithms and applications. The chapters are organized into three parts: Population-based and quantum search algorithms, Search algorithms for image and video processing, and Search algorithms for engineering applications.