PAPER

A Fast Block Matching Algorithm Based on Motion Vector Correlation and Integral Projections

Mohamed GHONEIM[†a)], ***Nonmember***, **Norimichi TSUMURA**[†], **Toshiya NAKAGUCHI**[†], **Takashi YAHAGI**[†], ***and*** **Yoichi MIYAKE**[†], ***Members***

SUMMARY The block based motion estimation technique is adopted by various video coding standards to reduce the temporal redundancy in video sequences. The core of that technique is the search algorithm implemented to find the location of the best matched block. Indeed, the full search algorithm is the most straightforward and optimal but computationally demanding search algorithm. Consequently, many fast and suboptimal search algorithms have been proposed. Reduction of the number of location being searched is the approach used to decrease the computational load of full search. In this paper, hybridization between an adaptive search algorithm and the full search algorithm is proposed. The adaptive search algorithm benefits from the correlation within spatial and temporal adjacent blocks. At the same time, a feature domain based matching criteria is used to reduce the complexity resulting from applying the pixel based conventional criteria. It is shown that the proposed algorithm produces good quality performance and requires less computational time compared with popular block matching algorithms.

key words: *video coding, motion estimation, adaptive block matching, integral projection*

1. Introduction

The high redundancy existing among the successive frames of a video sequence makes it possible to achieve high compression ratio in video coding. The temporal correlation is exploited using motion estimation techniques. Motion estimation [1] is the process of evaluating the movements between adjacent frames. Among various algorithms for motion estimation, the block-matching algorithm (BMA) is mostly used in the framework of generic coding due to its simplicity [2], [3].

In block-matching algorithm as shown in Fig. 1, the current frame is divided into square non-overlapping blocks of size $N \times M$ pixels. The coordinates (k, l) of its upper-left corner address each block. Then, the motion estimation procedure starts. The basic idea underlying the block motion estimation is to measure the shift, in x (δ_x) and y (δ_y) directions, that minimizes the block distortion measure (BDM) between the current block and the candidate blocks within a search area of size $(2w + 1) \times (2w + 1)$ pixels centred on the desired location in a reference frame.

The BDM is a positive and increasing function $g(x)$, it may be defined as:

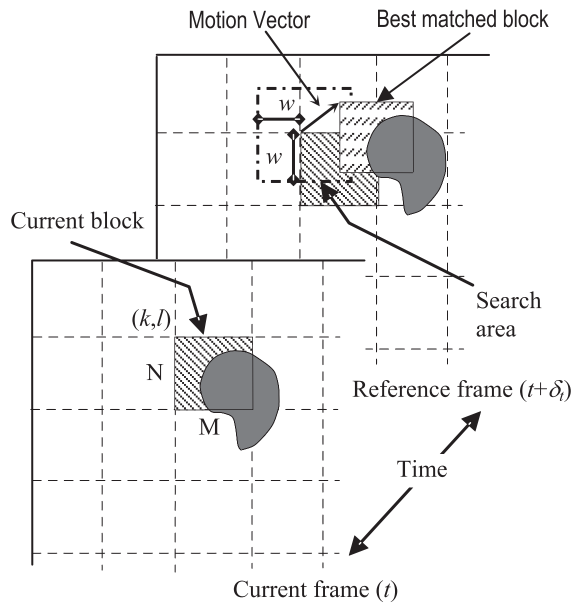


Fig. 1 Block matching algorithm.

$$BDM_{(k,l)}(\delta_x, \delta_y) = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} g(F_t(k+i, l+j) - F_{t+\delta t}(k+i+\delta_x, l+j+\delta_y)) \quad (1)$$

where $F_t(i, j)$ is the intensity of the pixel with coordinates (i, j) in frame at time t. If $g(x) = x^2$ then the BDM is called mean squared error (MSE), while, it is known as sum of absolute differences (SAD) if $g(x) = |x|$.

The goal of the search algorithm is to find $S^* = (\delta_x{}^*, \delta_y{}^*)$ that minimizes BDM, that is:

$$S^*(k, l) = \arg\min_{(\delta_x, \delta_y)} BDM_{(k,l)}(\delta_x, \delta_y). \quad (2)$$

By exhaustively testing all the candidate blocks within the search window, a full search (FS) algorithm gives the global optimum solution for the block motion estimation problem, but a substantial amount of computational load is demanded. To overcome this drawback, many fast but suboptimal block-matching algorithms have been developed.

The structure of this paper is as follows. In Sect. 2, a review of some fast BMAs is presented. Section 3 gives a detailed explanation for the proposed algorithm. In Sect. 4, experimental results for various sequences and discussions

Manuscript received July 14, 2008.
Manuscript revised October 15, 2008.
[†]The authors are with the Graduate School of Advanced Integration Science, Chiba University, Chiba-shi, 263–8522 Japan.
a) E-mail: mghoneim@graduate.chiba-u.jp
DOI: 10.1587/transinf.E92.D.310

of the results are given. Finally, Sect. 5 presents the conclusion for this research.

2. Review of Fast Block Matching Algorithms

The heavy computational load for a large search range is a significant problem in real-time video-coding applications. Therefore, the speedup problem makes it essential to introduce sub-optimal but faster search algorithms to reduce the computation demand of the full search.

The two-dimensional logarithmic search (TDLS) tracks the direction of minimum distortion, resulting in considerable computational simplicity. A modified algorithm [5] has been proposed for further reduction of computational complexity. Another famous technique called three-step search (TSS) [6] requires only 25 searching locations compared with the 225 required for full search when the searching range is defined as $w = 7$. Because of its simplicity and adaptation to hardware implementation a number of modifications have been done to the TSS. The new three-step search algorithm [7] considers a centre biased checking point pattern in the first search stage. Another modified TSS is the efficient TSS (ETSS) that employs a small diamond pattern in the first step, and the unrestricted search step is used to search the centre area [8].

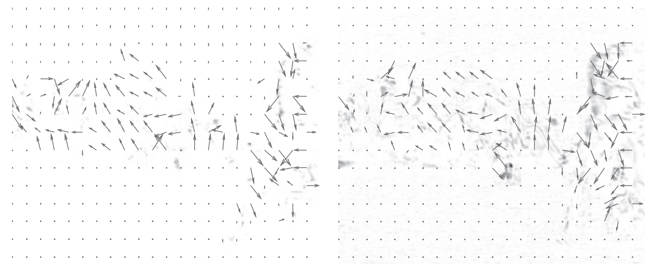
Puri et al. [9] introduced a hybrid of TDLS and TSS and called it orthogonal search algorithm (OS). The cross search algorithm, presented by Ghanbari [10] is very similar in many ways to TDLS, but the search pattern form a saltier cross sign (×) rather than Greek cross (+). The four-step search algorithm (4SS) [11] exploits the centre-biased characteristics of the real world video sequences by using a smaller initial step size compared with TSS. Tourapis et al. [12], [13] introduced the idea of zonal search algorithms that also depends on centre-biased characteristics of the real world video sequences.

The diamond search (DS) [14] employs two diamond-shape search patterns to find the location of best match block. It is implemented in the MPEG-4 video encoding environment and its efficiency is demonstrated through core experimental results [15]. Based on these results, it is adopted in MPEG-4 verification model [16].

From the previous review, it is clear that the computational complexity and quality performance for block motion estimation are contradictory. However, up to now, few algorithms could not only reduce the computational complexity but also improve the quality performance. Therefore that, the goal of this paper is to investigate an approach for achieving both the reduction of the computational time and good quality performance.

3. Proposed Algorithm

Neighbouring blocks are most likely belonging to the same moving object or to the static background. Considering this fact, the motion vector (MV) of the current block may be determined by examining the motion vector values of its neighbouring blocks. Figure 2 shows the motion vector field for two temporally adjacent frames accompanied with frame difference between original frame and compensated frame for "Football" sequence. By examining Figs. 2 (a) and (b) we can see that, blocks that are located within the same area (foreground or background) are highly correlated. Similarly, by examining Fig. 2 (a) or (b), we can notice the correlation between adjacent MVs within the same frame. The advantageous of this approach is two folded. Firstly, the reduction of number of search points required to find the best match block. Secondly, it helps on the correction of singular and erroneous motion vector. This approach is known as adaptive search strategy. In the following section, we well describe this strategy and well introduce our contributions toward achieving better performance.



a) Frame No. 2 b) Frame No. 3

Fig. 2 Motion vector temporal and spatial correlation.

3.1 Adaptive Search Strategy

Algorithms based on this strategy exploit the correlation between the current block and its neighboring blocks to adaptively predict an initial search center (ISC) other than the conventional zero motion vector. This adaptive initial search center (AISC) is obtained by calculating the statistical average (such as the mean, the median, etc.) of the neighboring MVs [17], [18] or selecting one of the neighboring MVs according to certain criteria [19]. On some cases, the search window's size, and pattern are redefined accordingly [20]. After finding the AISC, a search algorithm is performed starting at this new search center. It is clear that the selection of a proper searching algorithm is a key point in the performance of BMA.

Specifically speaking, in order to obtain an accurate MV prediction for the current block, two factors need to be considered: 1) a set that consists of the neighboring blocks whose MVs will be used to estimate the AISC, and 2) algorithm used for refining this initially estimated MV.

The proposed algorithm adopts this strategy to achieve good performance. Unlike the earlier algorithms, the proposed algorithm utilizes two improvements to increase the adaptive search strategy efficiency.

First, the proposed algorithm takes advantage of the correlation between adjacent motion vectors in the spatial and temporal domains simultaneously. That is, we propose to use a set of MVs from the current and previous motion

Table 1 MV probabilities distribution.

w	0	1	2	3	4	5	6	7
0	49.667	16.541	7.976	2.876	1.099	0.518	0.252	0.429
1	4.671	3.211	1.221	0.734	0.498	0.561	0.201	0.246
2	1.002	0.912	0.611	0.442	0.203	0.143	0.104	0.187
3	0.412	0.410	0.241	0.201	0.131	0.099	0.081	0.153
4	0.339	0.230	0.114	0.090	0.099	0.081	0.074	0.134
5	0.147	0.118	0.069	0.065	0.053	0.066	0.054	0.128
6	0.121	0.251	0.055	0.041	0.044	0.046	0.043	0.131
7	0.431	0.133	0.125	0.097	0.077	0.063	0.088	0.362

Table 2 Probabilities and cumulative probabilities for different patterns.

w	0	1	2	3	4	5	6	7	Sum.
P(SP)	49.66	24.42	11.72	5.32	2.80	1.92	1.37	2.78	100.0
CP(SP)	49.66	74.09	85.81	91.13	93.93	95.85	97.22	100.0	
P(DP)	49.66	21.21	12.19	5.42	3.19	2.08	1.57	1.75	97.07
CP(DP)	49.66	70.88	83.07	88.49	91.68	93.75	95.32	97.07	
P(CDP)	49.66	3.21	0.61	0.20	0.10	0.07	0.04	0.36	54.26
CP(CDP)	49.66	52.88	53.49	53.69	53.79	53.85	53.90	54.26	

fields. That set is called set of confidence (SOC). Although this consumes more memory to maintain the values of previous motion vector field, it adds more accuracy to the prediction of the AISC that leads to a great improvement in the correctness of estimated and final MV.

Furthermore, we studied the statistical distribution of MV for choosing the proper pattern of SOC. The percentage of MV probabilities distribution (MVPD) within a search region of width $w = 7$ pixels for 16×16 blocks computed from the first 100 frames of 16 video sequence are shown in Table 1. Moreover, Table 2 shows the percentage probabilities, P(.), and percentage cumu-lative probabilities, CP(.), of MV occurrence for square (SP), diamond (DP), cross diagonal (CDP) patterns. In conclusion, for search region of size $w = 1$, the SP has 74% of MVs, while for DP and CDP the motion vector occurrence are 70.88% and 52.88% respectively. On the other hand, SP consists of 9 blocks while DP and CDP are consisting of 5 blocks. Therefore, the proposed algorithm used the DP since it requires small number of blocks with higher accuracy.

Figure 3 depicts the SOC that may be defined as: SOC = $\{B1_{(m-1,n,t)}, B2_{(m,n-1,t)}, B3_{(m,n,t-1)}, B4_{(m+1,n,t-1)}, B5_{(m,n+1,t-1)}\}$ where $Bi_{(m,n,t)}$ is the block in location (m, n) on frame at time t. The first two blocks belong to the spatial domain (the same frame as the current block), while the other three blocks belong to temporal domain (the reference frame).

The second technique used to improve the performance of adaptive search strategy is to put more emphasis on blocks with lower BDM value, and less emphasis on blocks with higher BDM value while computing the AISC. That is, high confidence neighbouring blocks, guide that of low confidence. To achieve this goal, we define a weighting vector W = $\{w_B,\ B \in \text{SOC}\}$, such that the values of its component depend on the value of the BDM for each block in the SOC.

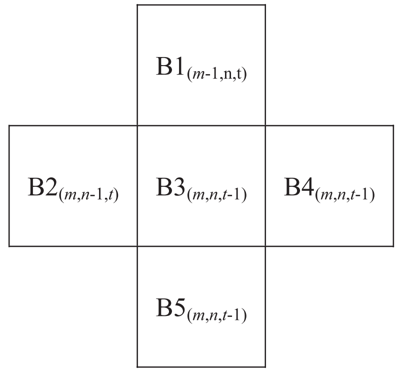


Fig. 3 SOC for predicting the adaptive initial search center.

The effect of the weighted MVs acts as adding a smoothening term to Eq. (2) as follows:

$$S^*(k,l) = \arg\min_{(\delta_x,\delta_y)}[BDM_{(k,l)}(\delta_x,\delta_y)] + \sum_{B\in SOC} w_B\vec{v}_B, \quad (3)$$

where $\vec{v}_B$ is the motion vector for the block B. This term enables the search area to flow with the moving object by selecting a proper initial search center. The major advantage is that it can increase the chance of finding the true motion vector and reduce the computational requirement. This advantage is clear for sequences with high motion activity since the BDM is significantly high, as well be shown in the simulation results.

On the other hand, selecting wrong candidate leads to searching in wrong direction. As a result, the search algorithm for best match block either consumes more computation time or is trapped into local minima. To overcome this disadvantage, the proposed algorithm implements the FS algorithm if the BDM for the estimated AISC (BDMI) is larger than a threshold value. Furthermore, if BDMI is lower than another threshold value, then the motion vector of the AISC is considered as the best MV and the search process stops.

To diminish the excess in the algorithm complexity upon using the FS algorithm, the proposed algorithm uses a feature domain based criteria. Subsequently, a minimal set of blocks is rechecked using conventional matching criteria to assure the accuracy of final MV. The following section introduces the feature domain criteria and the rechecking process. Moreover, it explains how it helps in improving the proposed algorithm performance.

3.2 Feature Domain Criteria

A feature domain based criterion is used to reduce the complexity of BMAs. Kim and Park [21] proposed a feature-based BMA using the concept of integral projection*s* (IP) that reduce the computation time by a factor of two on the average.

IP methods reduce the computations by transforming a block of $N \times M$ pixels into two $N \times 1$ and $1 \times M$ vectors, as shown in Fig. 4. The first vector is related to the rows while

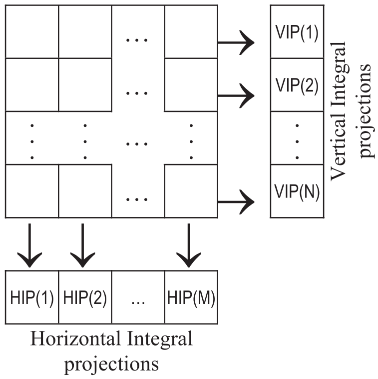


Fig. 4 Integral projections.

the other is related to the columns of the block. The *m*th component in the vertical projection vector is calculated by summing the pixels of the *m*th row, as follows

$$VIP_{t;k,l}(m) = \sum_{i=1}^{M} F_t(m, l+i) \tag{4}$$

The horizontal integral projection (HIP) is similarly computed.

After transforming the current block and the candidate blocks in the reference frame into its horizontal and vertical IP, the matching criteria at location (k, l) is given by the following formula:

$$D(k,l) = \sum_{r=1}^{N} \left|(HIP_{t;k,l}(r) - HIP_{t-1;k,l}(r))\right| + \sum_{c=1}^{M} \left|(VIP_{t;k,l}(c) - VIP_{t-1;k,l}(c))\right| \tag{5}$$

IPFS well be used to denote the FS algorithm using the integral projection matching criteria.

The proposed algorithm uses the IP features to select a minimal set (S) of candidate blocks. The elements of S are the blocks with the lowest IP matching criteria. Those candidate blocks are once more examined -rechecked- using the conventional matching criteria. The rechecking process ensures the improvement in compensated frame quality on the expense of the computations time. That is, the number of elements of S (N_S) is an important factor in the performance of the proposed algorithm. We tested different values of N_S, $N_S = 0^{\dagger}$, 3, 5, 7, and we found that the best value that satisfy the trade off between quality and complexity is $N_s = 5$.

3.3 Proposed Algorithm Procedure

For each frame, two thresholds T_1 and T_2 are calculated to divide the range of the previous frame BDM into three regions. The value of T_1 and T_2 dramatically affects the performance of the proposed algorithm. We tested a number of methods and we chose the best of them. The value of the thresholds is automatically computed for each frame depending on two statistical factors, namely the range and the mean value of BDM for the previous frame:

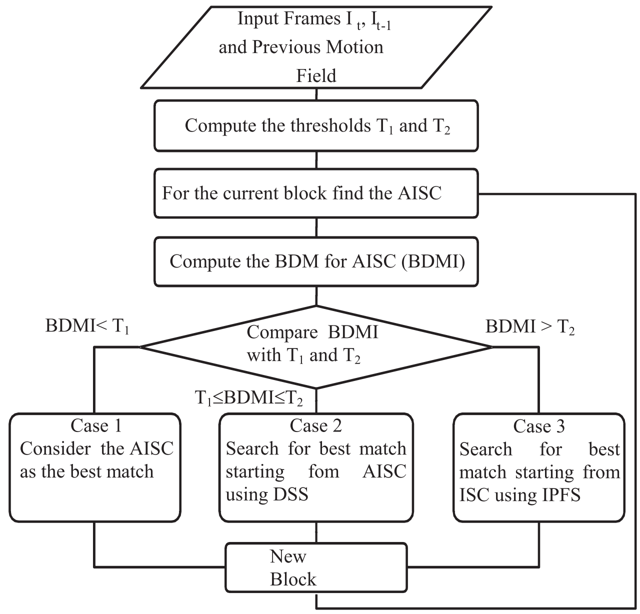


Fig. 5 Flow chart for the proposed algorithm.

$$T_1 = Min_{BDM} + \frac{Max_{BDM} - Min_{BDM}}{Mean_{BDM}}$$
$$T_2 = Max_{BDM} - \frac{Max_{BDM} - Min_{BDM}}{Mean_{BDM}}$$

where Max_{BDM}, Min_{BDM} and $Mean_{BDM}$ are the maximum, minimum and mean values for the previous frame BDM respectively.

The BDM calculated from the AICS (BDMI) has three cases

Case 1: BDMI $< T_1$,
Case 2: $T_1 \leq$ BDMI $\leq T_2$,
Case 3: BDMI $> T_2$.

In the first case, we well consider the motion vector of AISC as the best motion vector. That is, we obtain a hypothetical optimal motion vector (HOMV) by calculating BDM for one block only that decreases the computations time of the algorithm.

In the second case, the search process starts from the AISC applying one of the sub-optimal and fast search algorithms. We chose *dual stage search* (DSS) [22] algorithm as the sub-optimal search algorithm since it is proved to be fast and accurate enough to find the best motion vector.

Finally, in the third case, since the BDM for the AISC is high compared to BDM of all the blocks in the previous frame, the FS algorithm is applied to decrease the BDM. In this case, the search process starts from the ISC. Figure 5 depicts the flowchart of the proposed algorithm. The proposed algorithm is referred to as *hybrid adaptive dual stage search* (HADSS).

$^{\dagger}N_S = 0$ means no rechecking process is conducted.

4. Experimental Results and Discussion

In this section, we present the experimental results for the proposed algorithm. We compare FS, TSS, DS, DSS and adaptive rood pattern search (ARPS) [18] with HADSS in both quality of the compensated frame and computational complexity. A set of 16 sequences with different motion activity, size and format are used in the simulation. Table 3 shows all information for the test video sequences.

We used the Average MSE (A-MSE) and average PSNR (A-PSNR) between the original frame and the compensated frame to compare the quality performance of the algorithm. The average time (A-Time) elapsed per frame is used to measure algorithm complexity. Moreover, speed-up ratio (SUR) for each algorithm is used to figure out the improvement in algorithm complexity compared to FS as a reference algorithm. SUR is defined as:

$$SUR = \frac{Time_{FS}}{Time_{A}} \tag{6}$$

where $Time_{FS}$ and $Time_{A}$ are time elapsed by FS algorithm and the algorithm under investigation respectively.

The performance comparisons between different search algorithms in terms of speed-up, MSE, and PSNR for a number of sequences are illustrated in Tables 4, 5, 6 and 7, while others are shown in Figs. 8, 9 and 10. The FS algorithm is considered as a reference while computing differences in MSE (δ_{MSE}) and PSNR (δ_{PSNR}). That is

$$\delta_{MSE} = MSE_{FS} - MSA_{A} \text{ and } \delta_{PSNR} = PSNR_{FS} - PSNR_{A}$$

where MSE_{FS} and $PSNR_{FS}$ are MSE and PSNR for FS algorithm, while MSE_{A} and $PSNR_{A}$ are MSE and PSNR for algorithm under investigation.

It is noticed from the tables that, the proposed algorithm reduces the complexity of the FS by a factor of 10 to 12 with low degradation in PSNR ranging from 0.007 dB to 0.354 dB. In addition, the comparison of the complexity reduction with the other algorithms shows that our algorithm has low computational load.

For sequences with high motion activity, the proposed

Table 3 Sequences used in simulation.

Motion Activity	Sequence name (Format)
Low motion activity with stationary background and stable camera	Container (SIF), Akiyo (QCIF), Miss America (QCIF), news (QCIF), Sales Man (CIF), and Clair(CIF)
High motion activity with stationary back ground and stable camera	Foreman (QCIF), Hall Monitor (SIF), Suzie(QCIF), and Mother & Daughter (QCIF)
High motion activity with stationary back ground and stable camera	Car Phone (QCIF), Coast Guard(QCIF, SIF) and mobile (SIF, ITU)
Very High motion activity with moving background and camera zooming and panning	Football(ITU, CIF), Garden (SIF) and Tennis(SIF)

Table 4 Speed-up, MSE, and PSNR for "Football" sequence.

Algorithm	A-Time	SUR	A-MSE	δ_{MSE}	A-PSNR	δ_{PSNR}
FS	1.440	1.000	361.185	0.000	22.602	0.000
TSS	0.130	11.076	407.055	-45.871	22.083	0.520
DS	0.220	6.548	406.321	-45.136	22.095	0.508
ARPS	0.203	7.102	424.164	-62.980	21.914	0.689
DSS	0.127	11.338	451.510	-90.325	21.633	0.969
HADSS	0.131	10.962	392.554	-31.369	22.248	0.354

Table 5 Speed-up, MSE, and PSNR for "Coast Guard" sequence.

Algorithm	A-Time	SUR	A-MSE	δ_{MSE}	A-PSNR	δ_{PSNR}
FS	0.393	1.000	63.026	0.000	30.893	0.000
TSS	0.037	10.608	68.237	-5.211	30.761	0.132
DS	0.049	7.956	67.494	-4.468	30.813	0.080
ARPS	0.045	8.790	65.487	-2.461	30.791	0.102
DSS	0.034	11.634	71.198	-8.172	30.757	0.136
HADSS	0.034	11.693	65.098	-2.072	30.843	0.050

Table 6 Speed-up, MSE, and PSNR for "Hall Object" sequence.

Algorithm	A-Time	SUR	A-MSE	δ_{MSE}	A-PSNR	δ_{PSNR}
FS	0.373	1.000	20.971	0.000	35.767	0.000
TSS	0.037	10.162	21.050	-0.080	35.754	0.012
DS	0.045	8.350	20.974	-0.003	35.766	0.001
ARPS	0.038	9.868	21.982	-1.011	35.615	0.152
DSS	0.028	13.422	21.068	-0.098	35.754	0.013
HADSS	0.031	12.122	21.068	-0.098	35.754	0.013

Table 7 Speed-up, MSE, and PSNR for "Clair" sequence.

Algorithm	A-Time	SUR	A-MSE	δ_{MSE}	A-PSNR	δ_{PSNR}
FS	0.368	1.000	3.960	0.000	42.980	0.000
TSS	0.036	10.284	3.973	-0.014	42.974	0.006
DS	0.042	8.853	3.964	-0.004	42.977	0.003
ARPS	0.037	10.021	4.023	-0.064	42.937	0.043
DSS	0.028	13.289	3.971	-0.012	42.973	0.007
HADSS	0.031	12.015	3.971	-0.012	42.973	0.007

Table 8 SIR% and MSE% of HADSS compared to TSS, DS and DSS.

Motion	Sequence	Algorithm	SIR%	MSE%
Low activity	Miss America	TSS	13.531	-2.062
		DS	40.582	1.143
		DSS	-9.524	0.164
	Mother and Daughter	TSS	16.826	2.457
		DS	43.563	3.967
		DSS	-9.461	2.743
Moderate Activity	Coast Guard	TSS	10.228	-4.606
		DS	46.973	-3.549
		DSS	0.514	-8.570
High activity	Football	TSS	-1.036	-3.562
		DS	67.410	-3.393
		DSS	-3.322	-13.062
	Garden	TSS	12.473	-14.832
		DS	104.73	-0.118
		DSS	10.972	-2.394
	Tennis	TSS	9.85	-14.44
		DS	83.76	12.96
		DSS	-0.04	-1.20

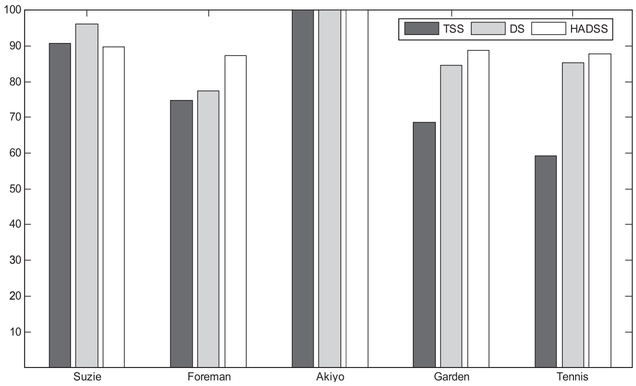


Fig. 6 Motion vector coincident probability.

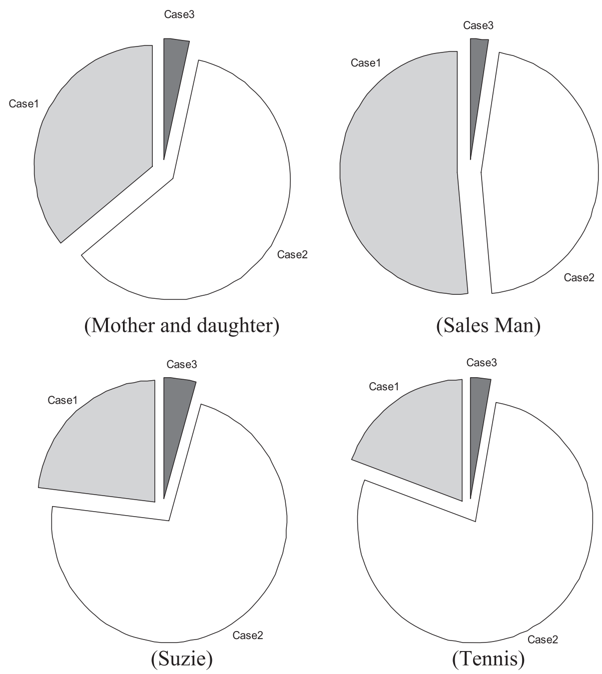


Fig. 7 Percentage of applying different cases.

algorithm has superior performance. For example, for "Football" sequence, HADSS has high SUR and the lowest MSE. In addition, the gain on PSNR is recognizable especially over DSS and ARPS algorithm. Meanwhile, for moderate motion activity sequence, such as "Coast Guard", the proposed algorithm is the fastest with the smallest distortion error. That is, the proposed algorithm not only achieves better quality as a gain in PSNR and reduction on MSE, but also it reduces the algorithm complexity.

On the other hand, for low motion activity such as "Hall Object" and "Clair", HADSS is more than 12 time faster than the FS with slight increase on MSE. However, for some sequences, degradations of quality produced by DS and TSS are lower than HADSS. Nevertheless, if we consider both the complexity reduction and the quality performance, we will notice that HADSS is superior to both algorithms. To emphasis this idea we make another comparison between the DS, TSS and HADSS in terms of speed improvement rate (SIR%) and percentage increase in MSE (MSE%) which may be defined as:

$$\mathrm{SIR}\% = \left(\frac{\mathrm{Time}_1 - \mathrm{Time}_2}{\mathrm{Time}_2}\right)100\%,$$

$$\mathrm{MSE}\% = \left(\frac{\mathrm{MSE}_1 - \mathrm{MSE}_2}{\mathrm{MSE}_2}\right)100\%,$$

where Time_1 and MSE_1 is the average time and MSE for HADSS, while, Time_2 and MSE_2 is the average time and MSE for DS, DSS or TSS. The result of that comparison is illustrated Table 8. This comparison shows that improvement in speedup, SIR% is positive, accompanied with increase in quality gain, MSE% is negative, prove that the proposed algorithm has superior performance specially for moderate and high motion activity sequences.

Furthermore, since the FS can produce the true motion vector, we define the motion vector coincident probability (MVCP) as a percentage ratio of motion vectors correctly found by various algorithms to that found by FS. Figure 6,

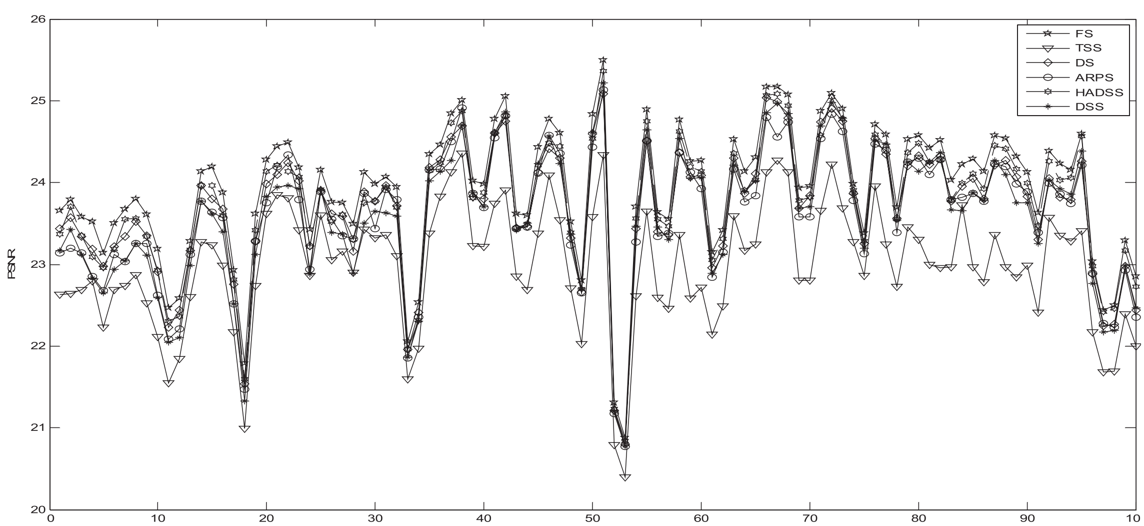


Fig. 8 MSE per frame for "Garden" Sequence.

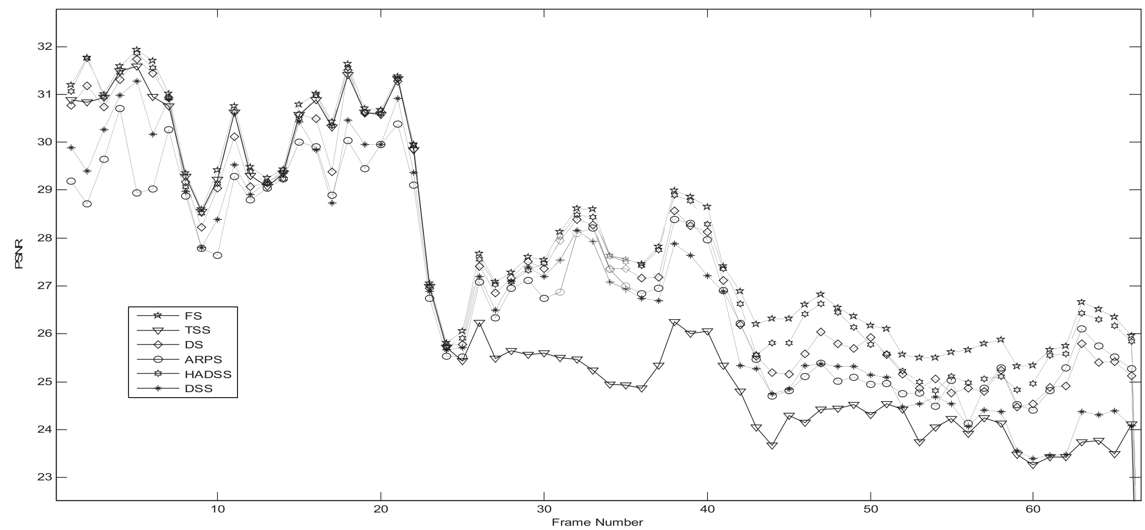


Fig. 9 MSE per frame for “Tennis” sequence.

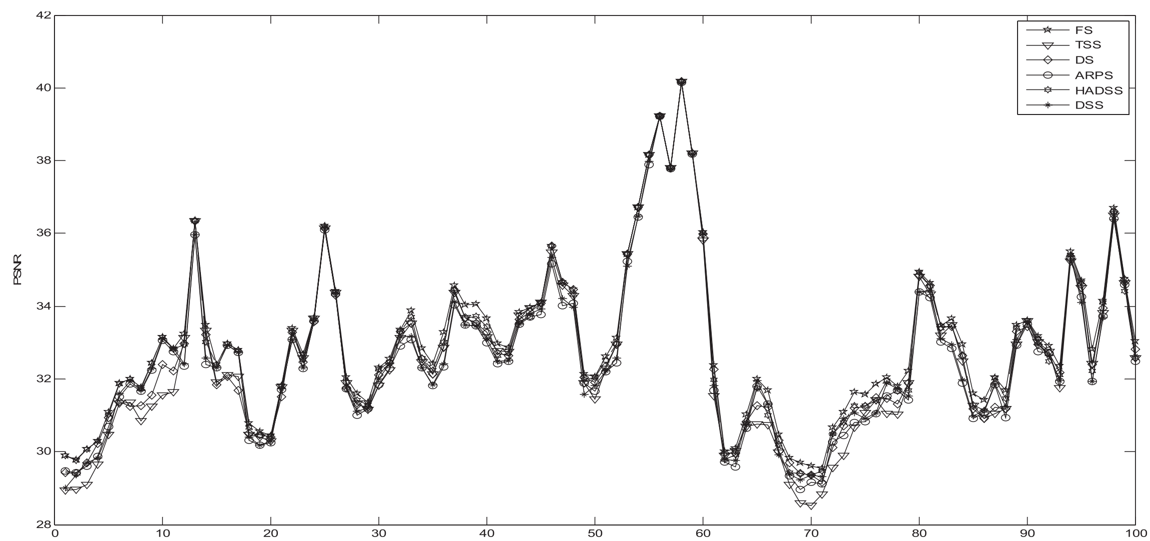


Fig. 10 PSNR per frame for “Foreman” sequence.

shows that the proposed algorithm has higher MVCP than TSS and DS except for “Suzie” sequence. That is, in general our algorithm is more accurate than other algorithms especially for sequences with high and moderate motion activity.

In Addition, Fig. 7 shows the percentage of applying the three cases for different video sequence. In general, Case 2 is the dominant case while Case 3 is the case with lowest occurrence since it is applied only for correcting motion vectors with large BDM. On the other hand, adopting the HOMV as the best match, ranges from 20% to 55% according to the motion activity that present on the video sequence.

Finally, From Figs. 8, 9 and 10, it is clear that all algorithm has almost the same performance in the areas of low motion activity, while in the high motion activity areas, the performance of the algorithms have a lot of deviations from the ideal algorithm (FS). Proposed algorithm has almost the minor deviation from the FS algorithm.

5. Conclusion

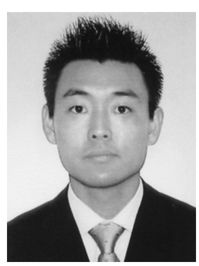
In this paper, we have presented a new fast algorithm for estimating motion in image sequence. Experimental results show that the proposed algorithm has a good performance compared to some well-known algorithms in terms of MSE, PSNR and computational requirement. Moreover, the performance of the proposed algorithm is superior to other algorithms for sequences with high motion activity due its adaptability of the proposed algorithm, which leads to avoiding local minima by tracking the major trends of the motion at the initial stage of searching procedure. Moreover, the gain on algorithm speed-up is due to HOMV obtained from the implementation of the first case.

References

[1] C. Stiller and J. Konrad, "Estimating motion in image sequences," IEEE Signal Process. Mag., vol.6, no.4, pp.70–91, July 1999.
[2] R. Steinmetz, "Data compression in multimedia computing: Standards and systems," Multimedia Syst., vol.1, pp.187–204, March 1994.
[3] F. Dufaus and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," Proc. IEEE, vol.83, pp.858–876, June 1995.
[4] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol.COM-29, no.12, pp.1799–1808, Dec. 1981.
[5] S. Kappagantula and K.R. Rao, "Motion compensated interframe image prediction," IEEE Trans. Commun., vol.COM-33, no.9, pp.1011–1014, Sept. 1985.
[6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," Proc. National Telecommunications Conference, vol.4, pp.G5.3.1–G5.3.5, 1981.
[7] R. Li, B. Zeng, and M.L. Liou, "A new three-step search algorithm for block motion estimation," IEE Trans. Circuits Syst. Video Technol., vol.4, pp.438–442, Oct. 1994.
[8] X. Jing and L-P. Chau, "An efficient three-step search algorithm for block motion estimation," IEEE Trans. Multimed., vol.6, no.3, pp.435–438, June 2004.
[9] A. Puri, H.-M. Hang, and D. Schilling, "An efficient block-matching algorithm for motion-compensated coding," IEEE International Conference on ICASSP '87, vol.12, pp.1063–1066, 1987.
[10] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans. Commun., vol.38, no.7, pp.950–953, July 1990.
[11] 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.4, no.3, pp.313–317, June 1996.
[12] A.M. Tourapis, O.C. Au, and M.L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol.12, no.10, pp.934–947, Sept. 2000.
[13] A.M. Tourapis, H.Y. Cheong, O.C. Au, and M.L. Liou, "N-Dimensional zonal algorithms. The future of block based motion estimation," IEEE International Conference on Image Processing (ICIP'01) Proceedings, Thessalonica, Greece, Oct. 2001.
[14] S. Zhu and 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.
[15] K.K. Ma and P.I. Hosur, "Core experimental results of fast motion estimation based on new test conditions (Q4a)," ISO/IEC JTC1/SC29/WG11/M4934, Vancouver, B.C., Canada, July 1999.
[16] "MPEG-4 video verification model, ver. 14.0," ISO/IEC JTC1/SC29/WG11/N2932, Oct. 1999.
[17] L.-J. Luo, C. Zou, and X.-Q. Gao, "A new prediction search algorithm for block motion estimation in video coding," IEEE Trans. Consum. Electron., vol.43, no.1, pp.56–61, Feb. 1997.
[18] K.K. Ma and P.I. Hosur, "Core experimental results of fast motion estimation based on new test conditions (Q4a)," ISO/IEC JTC1/SC29/WG11/M4934, Vancouver, B.C., Canada, July 1999.
[19] Y. Nie and K. Ma, "Adaptive rood pattern search for fast block matching motion estimation," IEEE Trans. Image Process., vol.11, no.12, pp.1442–1449, Dec. 2002.
[20] K. Venkatachalapathy and R. Kreshnamorthym, "A new adaptive search strategy for fast block based motion estimation algorithms," J. Vis. Commun. Image Represent., vol.15, pp.203–213, 2004.
[21] J.S. Kim and R.H. Park, "A fast feature-based block matching algorithm using integral projections," IEEE J. Sel. Areas Commun., vol.10, no.5, pp.968–871, June 1992.
[22] M. Ghoneim, A. Haggag, J. Lu, and T. Yahagi, "A very fast and efficient search algorithm for block motion estimation," J. Signal Process., vol.11, no.1, pp.69–74, Jan. 2007.

Mohamed Ghoneim received his B.Sc., and M.Sc., degrees from Mansoura University, Egypt, in May 1992, and December 1998, respectively. Both were in computer science. He was with Faculty of Science, Mansoura University from 1994 to 2005. In April 2005, he joined the Graduate School of Science and Technology of Chiba University, Japan, as a Ph.D. student. His current research interests are in the field of block motion estimation and its use in image and video coding.

Norimichi Tsumura was born in Wakayama, Japan, on 3rd April 1967. He received the B.E., M.E. and D.E. in applied physics from Osaka University in 1990, 1992 and 1995, respectively. He moved to the Department of Information and Image Sciences, Chiba University in April 1995, as assistant professor. He is currently associate professor since 2002, and also researcher at PREST, Japan Science and Technology Corporation (JST). He was visiting scientist in University of Rochester from March 1999 to January 2000. He got the Optics Prize for Young Scientists (The Optical Society of Japan) in 1995, and Applied Optics Prize for the excellent research and presentation (The Japan Society of Applied Optics) in 2000. He received the Charles E. Ives award in 2002 from the IS&T. He is interested in the color image processing, computer vision, computer graphics and biomedical optics.



Toshiya Nakaguchi was born in Kobe, Japan, on April, 1975. He received the B.E., M.E., and Ph.D. degrees from Sophia University, Tokyo, Japan in 1998, 2000, and 2003, respectively. He was a research fellow supported by Japan Society for the Promotion of Science from April 2001 to March 2003. Currently, he is an Assistant Professor of imaging science at the Graduate School of Advanced Integration Science, Chiba University, Chiba Japan, and is a research fellow in Center of Excellence in Visceral Biomechanics and Pain supported by CIRIUS, Danish Ministry of Education from September 2006 to May 2007. His current research interests include the computer assisted surgery and medical training, medical image analysis, real-time image processing, and image quality evaluation.

Takashi Yahagi received his B.S., M.S., and Ph.D. degrees all in electronics from the Tokyo Institute of Technology, Tokyo, Japan, in 1966, 1968 and 1971, respectively. In 1971, he joined Chiba University, Chiba, Japan, as a Lecturer in the Department of Electronics. From 1974 to 1984, he was an Associate Professor. In 1984, he was promoted to Professor in the Department of Electrical Engineering. From 1989 to 1998, he was with the Department of Information and Computer Sciences. Since 1998, he has been with the Graduate School of Science and Technology, Chiba University. His current research interests are in the theory and applications of digital signal processing and other related areas. He is the author of "Theory of Digital Signal Processing," Vols.1–3 (1985, 1985, 1986), "Digital Signal Processing and Basic Theory" (1996), "Digital Filters and Signal Processing" (2001), "Kalman filter and adaptive signal processing" (2005) and the co-author of "Digital Signal Processing of Speech and Images" (1996), "VLSI and Digital Signal Processing" (1997), "Multimedia and Digital Signal Processing" (1997), "Neural Network and Fuzzy Signal Processing" (1998): "Communications and Digital Signal Processing" (1999), "Fast Algorithms and Parallel Signal Processing" (2000) (Corona Pub., Tokyo, Japan). He is the Editor of the "Library of Digital Signal Processing" (Corona Pub., Tokyo, Japan). Since 1997, he has been the President of the Research Institute of Signal Processing, Japan, and also the Editor-in-Chief of the Journal of Signal Processing. Prof. Yahagi is a Senior Member of IEEE (USA) and a member of the Research Institute of Signal Processing Japan.

Yoichi Miyake has been professor of Chiba University since 1989. He received Ph.D. from Tokyo Institute of Technology in 1978. During 1978 and 1979, he was a post doctoral fellow of Swiss Federal Institute of Technology (ETHZ). In 1997, he was a guest professor of University of Rochester. He received Charles E Ives Award (paper award) from IS&T in 1991, 2001 and 2005. He became a fellow of IS&T in 1995. He was named as Electronic Imaging Honoree of the year in 2000 from SPIE and IS&T. He became honorary member of IS&T in 2003. He published many books and original papers on the image processing, color science, image evaluations and he is known as a pioneer of spectral image processing. He was served as a president of SPSTJ (The Society of Photographic Science and Technology of Japan) from 2000 to 2002 and a vice president of IS&T (The Society for Imaging Science and Technology, USA) from 2000 to 2004. He was also served as a president of The Japanese Association of Forensic Science and Technology from 1998 to 1999. He is currently served as professor and director of Research Center for Frontier Medical Engineering in Chiba University.