Proceedings of ITC-CSCC 2000, Pusan, Korea

Fast and Efficient Search Algorithm of Block Motion Estimation

Sang- Gyoo Kim, Tae-Ho Lee, Tae-Yeon Jung, and Duk-Gyoo Kim

School of Electronic and Electrical Engineering, Kyungoook National University,
1370 San-Kyuk Dong, Puk Gu, Tae Gu, 702-701, Korea
Tel: +82-53-950-5525, Fax: +82-53-950-5505
E-mail: sang--kyu@palgong.kyungpook.ac.kr

Abstract: Among the previous searching methods, there are the typical methods such as full search and three-step search, etc. Block motion estimation using exhaustive search is too computationally intensive. To apply in practice, recently proposed fast algorithms have been focused on reducing the computational complexity by limiting the number of searching points. According to the reduction of searching points, the quality performance is aggravated in those algorithms. In this paper, We present a fast and efficient search algorithm for block motion estimation that produces better quality performance and less computational time compared with a three-step search (TSS). Previously the proposed Two Step Search Algorithm (TWSS) by Fang-Hsuan Cheng and San-Nan sun is based on the ideas of dithering pattern for pixel decimation using a part of a block pixels for BMA (Block Matching Algorithm) and multi-candidate to compensate quality performance with several locations. This method has good quality performance at slow moving images, but has bad quality performance at fast moving images. To resolve this problem, the proposed algorithm in this paper considers spatial and temporal correlation using neighbor and previous blocks to improve quality performance. This performance uses neighbor motion vectors and previous motion vectors in addition, thus it needs more searching points. To compensate this weakness, the proposed algorithm uses statistical character of dithering matrix. The proposed algorithm is superior to TWSS in quality performance and has similar computational complexity.

Index terms—Kept winner, pixel-decimation, mean-absolute difference (MAD).

I. Introduction

The coding of video sequences has been the focus of a great deal of research in resent years. High–definition television (HDTV), video conferencing and CD-ROM archiving are some of the better known applications. In video coding inter-frame prediction can be improved by motion estimation. If the motion of a pixel or block of pixels can be reasonably estimated between successive frames, the inter-frame prediction accuracy is improved. Both pixel recursive algorithms[1] and block matching algorithm had been developed. The former relates to motion estimation of individual pixels, whereas the latter deals with the motion of block of pixels. The pixel recursive algorithm is computationally intensive and has been seldom used in practice, even though it is very effective. On the other hand, BMA's in spite of the inherent assumption that all the pixels in the block have the same motion, have been utilized in practice. BMA's have been widely used in video coding standards. The full search algorithm (FSA), which exhaustively searches for the best matched block within the searching window, can get the optimal solution. Various fast algorithms that minimize the complexity, but at the same time degrade the quality, have been developed. Such as TSS[2], conjugate direction search (CDS)[3]. From the previous research, it is worth nothing that the computational complexity and quality performance for block motion estimation are contradictory. Up to now , few algorithms could not only reduce the computational complexity but also improve the quality performance when compared to TSS. TWSS[4] algorithm could not only reduce the computational complexity but also improve the quality performance, when compared to TSS. But the quality performance is aggravated in TWSS, when compared to the fast moving images like football sequences. Then the fast and efficient search algorithm of block motion estimation (FES) is proposed for block motion estimation to compensate the drawback of TWSS.

II. Overview of the TWSS

The ideas for TWSS are dithering pattern for pixel decimation and multiple candidates for pixel-decimation-based full search.

A. Dithering Pattern for Pixel Decimation

When matching a block from the present frame with that from the previous frame, the matching criterion is usually evaluated by using all the pixels in the block. Block matching is based on the assumption that all pixels in a block move by the same motion vector. If too few pixel of block are used, there will be a reduction in the accuracy of the block motion estimation. However, if too many pixels of a block are used, there will be an extra cost on computational time of the block motion estimation. In order to find a suitable number of sub-sampled pixels used to compute the block difference. We first define a pattern that is ordered dithering matrix D^N of size N × N. The elements of the dithering matrix denoted as $D^N_{x,y}$ have their integer numbers from zero to N^2-1, Where x and y mean the horizontal and vertical position of the dithering matrix. The number in the dithering matrix denotes the order to pick up the pixel in the block of NxN pixels. Fig.2 shows an example of a matching block of size 16 × 16 (the example of dithering matrix, $D^N_{x,y}$). The number in the block is

	1							8								16
	0	128	32	160	8	136	40	168	2	130	34	162	10	138	42	170
	192	64	224	96	200	72	232	104	194	66	226	98	202	74	234	106
	48	176	16	144	56	184	24	152	50	178	18	146	58	186	26	154
	240	112	208	80	248	120	216	88	242	114	210	82	250	122	218	90
	12	140	44	172	4	132	36	164	14	142	46	174	6	134	38	166
	204	76	236	108	196	68	228	100	206	78	238	110	198	70	230	102
	60	188	28	156	52	180	20	148	62	190	30	158	54	182	22	150
8	252	124	220	92	244	116	212	84	254	230	222	94	246	118	214	86
	3	131	35	162	11	139	43	171	1	129	33	161	9	137	41	169
y	195	67	227	99	203	75	235	107	193	65	225	97	201	73	233	105
	51	179	19	147	59	187	27	155	49	177	17	145	57	185	25	153
	243	115	211	83	251	123	219	91	241	113	209	81	249	121	217	89
	15	143	47	175	7	135	39	167	13	141	45	173	5	133	37	165
	207	79	239	111	199	71	231	103	205	77	237	109	197	69	229	101
	63	191	31	159	55	183	23	151	61	189	29	157	53	181	21	149
16	255	127	223	95	247	119	215	87	253	125	221	93	245	117	213	85

Fig.1. Pattern of the pixels used for computing the matching criterion.

numbered with the sequence of ordered dither matrix. Then we define the threshold T_l and T_h, where T_l and T_h are the number of sub-sampled, respectively. T_h-T_l+1 is the number of sub-sampled pixels in the block. The cost function $C_{i,\ j}(T_l\ ,\ T_h)$ of the block is then defined as follows:

$$C_{i,j}(T_l, T_h) = \{1/(T_h - T_l + 1)\} \sum_{D^N_{x,y}}^{T_h} g(f_n(x,y) - f_{n-1}(x+i, y+j))$$

Where $T_l \leq D^N_{x,y} \leq T_h$

From the above equation, $C_{i,j}(T_l\ ,\ T_h)$ is the cost function of the block, where g(•) is the distortion function. We can define $g(x)=x^2$ or $g(x)=|x|$. For the best match from Cost function $C_{i,j}(T_l\ ,\ T_h)$, we can find the motion vector $v_{i,j}$ of the block, where i, j denotes the displacement of the motion and the size of the searching window is ± 7. From the experimental results of three typical video sequences, it is shown that a low bound of 48 sub-sampled pixels indeed exists.

B. Multiple Candidates for Pixel-decimation-based full search

The MSE of pixel-decimation-based full search is larger than that of all-pixel-based full search. The method for overcoming this drawback is to keep more than one winner for the pixel-decimation-based full search. The tradeoff between computational time and accuracy is determined by experimental results and the kept winners are selected as the candidates with the least distortion. We can observe the relationship between the number of kept winners and the improvement index from the experiments. From the experimental result, it is shown that keeping six winners in each step is appropriate. And it is shown in Fig.2.

C. Description of TWSS

The TWSS combines the ideas of Ⅱ. A and Ⅱ. B. In Fig. 3, "○" means the searching points in first step. Among the searching points in first step, the six candidates with the least distortion are chosen by using 48 pixels(Ⅱ.A).

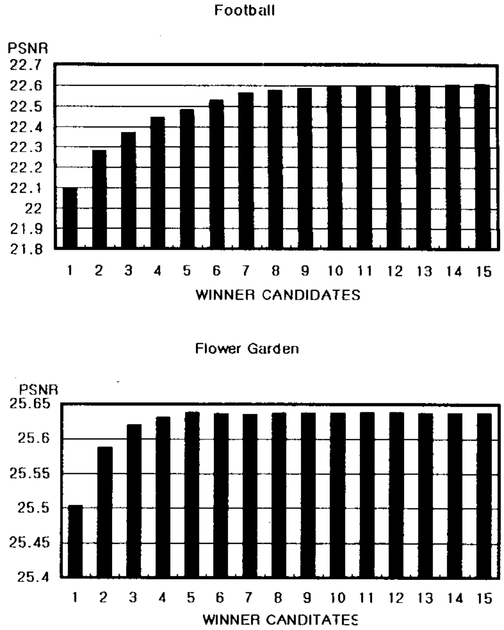

Fig.2. The improvement index of the football, flower garden sequences with difference number of kept winners.

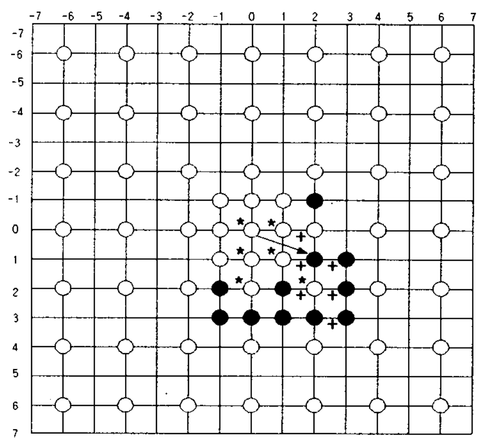

Fig.3. An Example of the TWSS for a searching range.

Among the six candidates with the least distortion and the eight neighboring points of the six candidates with the least distortion respectively, using 48 pixels in the second step again chooses the six candidates with the least distortion. At the final step, the candidate with the least distortion is chosen by used 256 pixels (like BMA's). This position of chosen candidate is used as motion vector in this block. Fig.3 shows a demonstrative example of the TWSS for a searching of ±7 pixels, whereas (i, j) is the vector of the motion. The pixels with " *" are the optimal points of the first step, and the pixels with "+" are the optimal points of the second step. The symbol "→" is denoted as the motion vector of the block, where "○" means the searching points of the first step and "•" means the searching points of the second step.

Ⅲ. The proposed Algorithm

The ideas for proposed algorithm (FES) are TWSS, temporal and spatial correlation and dithering matrix character to decrease computational complexity.

A. Temporal and spatial correlation

In this section, the motion of camera and objects in the previous frame and the correlation in moving objects in the current frame are considered. The motion of objects does not move in only a block but move with objects of neighbor blocks. In addition, the motion of objects in current block has high correlation with neighbor blocks at previous and current frames. For that reason, the motion vector of current block is predicted by the average motion vector of previous neighboring nine blocks like equation (1).

$$P\_MV(i,\ j)_k = (1/9)\sum_{n=-1}^{1}\sum_{m=-1}^{1} MV(i+n, \mathrm{j}+\mathrm{m})_{\mathrm{k}-1} \quad (1)$$

At motion vector $P\_MV(i, \mathrm{j})_{\mathrm{k}}$, the motion vector of current block may be predicted. $MV(i, \mathrm{j})_{\mathrm{k}-1}$ is motion vector of (k-1)'th frame. Thus, we acquire a motion vector to predict the motion vector of current block in this equation (1). And the other three-motion vector is acquired by prediction below. The motion of current block has high correlation with that of neighbor three blocks like Fig.4 in current frame.

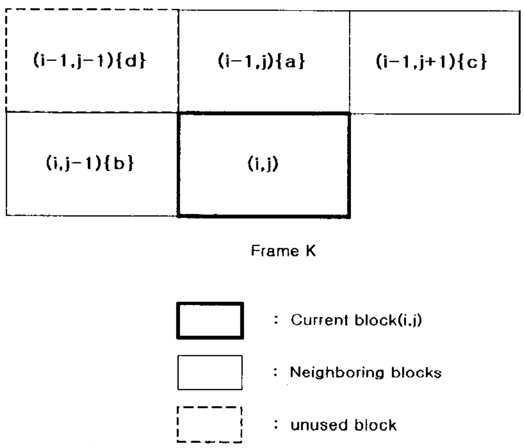

Fig.4. Neighboring blocks to estimate the motion vector of current block.

Those probabilities are very high in the frames. The motion vector of current block is predicted by motion vector of neighboring three blocks. In Fig .4, three blocks are presented as vertical block{a}, horizontal block{b}, and 45° block{c}. Because 135° block{d} has very high correlation with vertical and horizontal blocks, it is not used. In view of the result above achieved, the motion vector of current block is very likely to be close to the motion vector of the neighboring three blocks (a intra-frame concerned) and average motion vector of previous neighboring nine blocks (a inter-frame concerned).

B. Description of the Technique

The proposed algorithm called the FES (Fast and Efficient Search Algorithm of block motion estimation). Like the TWSS algorithm, FES adapts two-step search but FES considers spatial and temporal correlation using neighbor and previous block to improve quality performance. In Ⅲ.A, average one motion vector ($P\_MV(i,\ j)$) is acquired by using temporal correlation. And the three-motion vector of blocks ({a}, {b}, {c}) is acquired by using spatial correlation in Fig.4. These four motion vectors are used to compensate TWSS at the fast moving images. Assume that the searching range is defined as $\pm w$ pixels. Fig.3 shows the typical searching pattern of the first step with step size two but we must not consider black circles (●) in the first step, and the searching range is arranged as ±7 pixels, where i and j are horizontal and vertical directions of the searching window, respectively. Suppose that TS is defined as the number of sub-sampled pixels, within the block and TC is defined as the number of the kept winner. S is defined as the set of all candidate locations for the first step. M is defined as the set of winners kept for the first step and consists of the candidate locations for the second step, where m is the number of elements in M. N is defined as the set of winners kept for the second step, where n is the number of elements in N. $A(\ i,\ j)$ is defined as the set of neighbor locations of $(\ i,\ j)$. P is defined as the set of four motion vectors to compensate TWSS, where p is the number of elements in P. So to speak, P is defined as average motion vector of previous neighbor nine block ($P\_MV(i,\ j)$) and motion vectors of current neighbor block({a}, {b}, {c}). Then the FES algorithm can be described as follows:

Initial step: S = $\{(i\ ,j) \mid -w \le i\ ,j \le w$ and $i\ ,j$ is even$\}$ U $\{(\mathrm{i}\ ,j) \mid -1 \le i\ ,j \le 1\}$.
    M=φ, N=φ, P={ $P\_M.V(i, j)$, {a}, {b}, {c}}.
    $A(i,j) = \{(i-1, j-1), (\ i-1, j), (\ i-1, j+1), (i, j-1), (i,j+1), (i+1,j\text{-}1), (i+1,j), (i+1, j+1)\}$.
Step 1: For m=1 to TC do
    For all the $(\ i\ ,j) \in$ S, find $(i\ ,j)$
        such that $C_{i,j}(0,\ TS\text{-}1)$ is minimum.
        M={$(i\ ,j)$} U M. S=S-{$(i\ ,j)$}.
    End for.
Step 2: For all the $(\ i\ ,j) \in$ M do
        M = $A(i\ ,j)$ U M.
    End for
    For n=1 to TC do
    For all the $(\ i\ ,j) \in$ M, find $(i\ ,j)$
        such that $C_{i,j}(0, \mathrm{TS}\text{-}1)$ is minimum.
        N={$(i\ ,j)$} U N. M=M-{(i, j)}.
    End for.
        N=N U P
Final Step : For all the $(i\ ,j) \in$ N, find(i, j) such that
        $C_{i,j}(0, (N/2)^2\text{-}1)$ is minimum.

At final step of proposed algorithm because of using motion vectors (the proposed four motion vector at Ⅲ.A) to compensate motion at the fast moving images, the proposed algorithm makes use of the number of

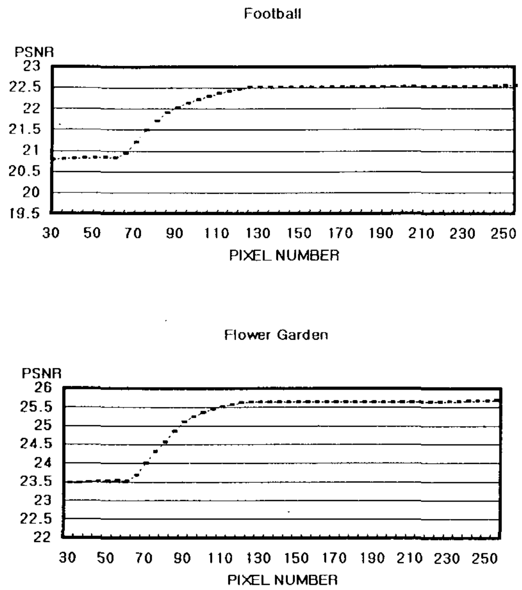

Fig.5. The statistical character of dithering matrix at final step.

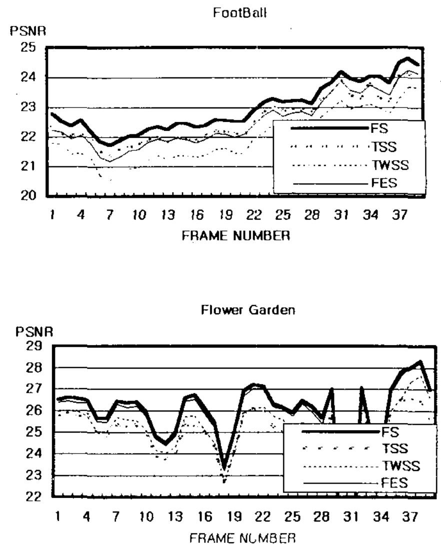

Fig.6. The PSNR comparisons of FSA, TSS, TWSS and FES(PSNR/Frame).

searching points more than TWSS. So the computational complexity is increased. To compensate this drawback, the proposed algorithm makes use of the statistical character of dithering matrix. Dithering matrix character is shown Fig.5. From the experimental results of typical video sequences, it is shown that a low bound of 125 sub-sampled pixels indeed exists. This used threshold value is not 256 pixels like TWSS but optimal 125 pixels by statistical character of dithering matrix. In final step, Though 125 pixels are used, PSNR is similar to the PSNR of using 256 pixels. Finally, first column blocks and first row blocks do not consider spatial and temporal correlation in frames, because it is border.

Ⅳ. Simulation and Conclusion

In this section, we show the experimental results of our proposed algorithm. We compare FSA, TSS and TWSS to FES in both quality performance and computational complexity. Three tested sequences, which are CCIR601 format of football, flower garden are used in the experiments. The sequence of football, flower garden, and table tennis is 40 frames respectively. In our experiment, the cost function for the block is defined as Mean Absolute Difference (MAD). The block size is fixed at 16×16 and the maximum in raw and column is assumed to be ± 7 pixels. Only the luminance component (Y component) in each frame is utilized in the experiment. The average PSNR comparisons of FSA, TSS, TWSS and FES are shown in Fig.6 and Table.1. The computational complexity per block (C/B) comparisons of FSA, TSS, TWSS and FES are shown in Table.2. In the result of experiment, we see that FES is superior to TWSS in quality performance and computational complexity is similar to TWSS for the every moving image.

Algorithms	PSNR(dB)		
	Football	Flower Garden	Table Tennis
FSA	22.9	25.8	26.2
TSS	22.5	24.8	25.4
TWSS	21.9	25.0	25.8
FES(proposed)	22.5	25.6	26.0

Table.1. The average PSNR comparisons of FSA, TSS, TWSS and FES.

Algorithms	Computational Complexity per Block (C/B)		
	Football	Flower Garden	Table Tennis
FSA	57600	57600	57600
TSS	6400	6400	6400
TWSS	4896	5136	5568
FES(proposed)	4860	5100	5657

Table.2. The computational complexity per block (C/B) comparisons of FSA, TSS, TWSS and FES.

References

[1] Y. Huang and X. Zhuang, "An adaptive refined block matching algorithm for motion compensated video coding." *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 56-59, Feb. 1995.

[2] T. Koga. et al., "Motion compensated interframe coding for video conferencing, " in *Proc. Nat.Telecommun*, LA, Nov. 13, 1981, pp, G5.3.1-5.3.3.

[3] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, pp. 950-953, July 1990.

[4] F. H. Cheng and S.N. Sun, "Nes Fast and Efficient Two-Step Search algorithm for block motion estimation" *IEEE trans. Circuits and Syst. Video Technol*, vol 9, no.7, Oct 1999, pp. 979-983.