An Improved Three-Step Search Algorithm for Block Motion Estimation

Won-Gi Hong*, Tae-Myoung Oh*, Sung-Jea Ko** *Regular Members*

ABSTRACT

The three-step search (TSS) algorithm for block motion estimation has been widely used in real-time video coding due to the simplicity of the algorithm, significant reduction of computational cost, and good performance. In this paper, an improved three-step search (ITSS) algorithm is proposed to improve the performance of the TSS algorithm. Simulation results show that, in terms of motion compensation errors, the proposed ITSS outperforms some popular fast search algorithms while it has the lower computational complexity.

I. Introduction

In many block-based video coding standards [1]-[3], motion estimation is a key technique for eliminating temporal redundancy of successive frames. The most simple and powerful method among many block-matching algorithms (BMAs) for motion estimation is the full search (FS) algorithm. However, this method is not suitable for real-time video applications due to its massive computational cost. To reduce the computation of the FS, many fast search algorithms have been proposed [4]-[8]. Among them, the three-step search (TSS) [4] is one of the most popular algorithms since it has advantages in its simplicity, low computations, and good performance.

It has been shown that more than 80% of blocks are stationary or quasi-stationary (within ± 1) even in the fast-motion "Football" sequence [8]. The TSS can be further improved based on the center-biased characteristics of the real-world image sequences. The new three-step search (NTSS) algorithm [6] introduces a center-biased search points pattern to the TSS to improve the performance. The four-step search (4SS) [7] and the unrestricted center-biased diamond search (UCBDS) [8] have shown better performance by exploiting the center-biased characteristics of image sequences. Although these algorithms perform better than the TSS, they require more computations than the TSS for finding the large motion vectors [6]-[8].

In this paper, we propose a new methodology for block motion estimation, called the improved TSS (ITSS). The proposed method, unlike the TSS, employs 13 search points including 4 extra center-biased search points in order to find small motion vectors efficiently. Moreover, a pixel subsampling scheme is incorporated for first-step block matching, significantly reducing the computation complexity without degrading the performance. It will be shown that the ITSS algorithm performs better than some popular fast algorithms in terms of motion compensation errors, computational complexity, and robustness.

II. Proposed Algorithm

1. Block Matching Criterion

In the block-based video coding, block motion estimation is carried out between the current frame and the previous frame of a video sequence. Each frame is divided into nonover-

* 명지전문대학 전자과 ** 고려대학교 전자공학과
논문번호 : 00108-0323, 접수일자 : 2000년 3월 23일

lapped blocks with size $M\times N$. If the search range is assumed W, the search area of the previous frame has the size $(2W+M)\times(2W+N)$. In general, the mean absolute difference (MAD) is popularly used as the block matching criterion due to its less computation complexity. The MAD can be expressed as

$$MAD(x,y)=\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} | S_k(p+i,q+j)-S_{k-1}(p+x+i,q+y+j)| \tag{1}$$

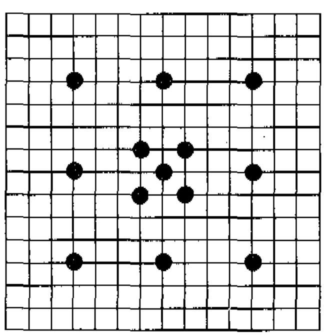
where $S_k(\cdot)$ and $S_{k-1}(\cdot)$, respectively, are pixel values in the *k*th and *k-1*th frame, and *(p,q)* denotes the coordinates of the upper left corner pixel of the block of the *k*th frame. If the minimum MAD is found at the location of *(x,y)* in the range of $-W\leq x,\ y\leq +W$, the *(x,y)* is determined as the motion vector.

2. Proposed ITSS Algorithm

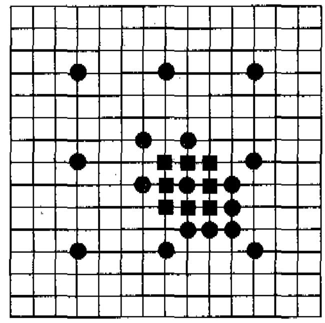
Unlike the TSS, the ITSS employs the initial center-biased search pattern to exploit the characteristics of the center-biased motion vectors. Fig. 1.(a) shows the 13 initial search points for the ITSS. In order to reduce the computational complexity of first-step block matching when the block size is 16×16, the MAD is computed at each point of the 2 to 1 subsampled block defined as

$$\{(x,y)\mid x+y=2k, k\in Z\} \tag{2}$$

where (x,y) is the location of a subsampled pixel of the block, and k is an integer.



(a)



(b)

Fig. 1 Initial search pattern and the next search strategy. (a) Initial 13 search points in the first step, (b) The next search path when the minimum in the first step is one of five center-biased locations.

First, we choose two candidate points with the first and second minimum MAD's among 13 initial search points. To obtain the accurate first-step minimum MAD, we compute new MAD's using all the pixels of the block at the two candidate points. If the first-step minimum MAD point is one of five center-biased search points, the search in the next step employs a halfway-stop technique using eight points neighbouring the minimum MAD point as shown in Fig. 1(b). If the center point becomes the minimum MAD point, the search stops. Otherwise, the search resumes using the previous minimum point plus neighbouring eight points. Thus, we can find the minimum MAD point within the central search area of 7×7. If the first-step minimum MAD point becomes one of outside 8 points, the next search will be the same as the TSS.

The detail of the proposed algorithm is presented below:

- **Initial Search:**
- Compare the MAD's obtained at 13 search points using subsampled 128 pixels of the block, and find two minimum candidate points.
- Among these two points, a first-step MAD point is determined by comparing new MAD's obtained using all the pixels of the block.
- If the determined point becomes one of five points in the central area, then go to Inside

Search. Otherwise, go to Outside Search.

· **Inside Search:**

- Check the candidate point plus neighbouring eight points as shown in Fig.1 (b) to find the minimum MAD point.
- If the center point becomes the minimum, the search stops. Otherwise, retry to check the minimum plus neighbouring eight points as shown in Fig. 1(b), and stop.

· **Outside Search:**

- The search in this step is the same as the TSS.

3. Analysis of the Computational Cost

In this subsection, we compute the number of search points per block for each step of the ITSS.

Although the initial search performs at 13 points, the actual number of search points for the initial search is equal to 7.5, since initial block matching using two to one subsampled pixel requires 6.5 (13/2) and the computation for determining the true minimum point among two minimum candidate points requires 1 (2/2). If the inside search is selected, additional 8 points are required. If the center point is the minimum in the inside search, the total number of search points equals 15.5 (7.5+8). Otherwise, the total number of search points varies from 18.5 (15.5+3) to 20.5 (15.5+5) depending on whether the previous minimum point is located at the corner of the search window or not. If the outside search is selected, the total number of search points is equal to 23.5 (7.5+8+8). Thus, the total number of search points of the ITSS is between 15.5 and 23.5, which is smaller than that of the TSS (fixed 25), NTSS (17 to 33), and 4SS (17 to 27) [4],[6],[7].

Ⅲ. Simulation Results

For our simulation, four sequences (Claire, Salesman, Football and Tennis) with different types of motion were used. The size of each frame is 352×240 pixels quantized to 8 bits. The block size and the search range used for our simulation are $M\times N = 16\times16$ and $W = 7$, respectively. First 90 frames of each sequence were used for the performance comparison of the FS, TSS, NTSS, 4SS, and ITSS. The mean absolute difference is used as the block matching criterion

Table 1. shows the performance comparison of aforementioned five search methods in terms of average MSE (mean square errors) between the original frame and the motion compensated frame. It is seen that the ITSS performs nearly as well as the NTSS for the small-motion sequences (Claire and Salesman) and outperforms the TSS, 4SS, and NTSS for the fast-motion sequences (Football and Tennis). The comparison of the average search points per motion vector estimation is presented in Table 2. It is seen that the proposed algorithm has a lower computational complexity than the other three fast algorithms. Table 3 shows that the ITSS achieves good results for the probability to find the true motion vector obtained by the FS algorithm. Especially, for the performance comparison for each frame, Fig. 3 shows the comparison of average search points for the first 90 frames for the Football and Tennis sequences.

Table 1. Average MSE of the First 90 Frames

Search Algorithm	Claire	Salesman	Football	Tennis
FS	5.41	21.03	180.84	178.70
TSS	5.54	21.95	202.18	223.81
NTSS	5.43	21.24	192.23	207.32
4SS	5.50	21.83	197.14	202.75
ITSS	5.43	21.67	189.44	185.96

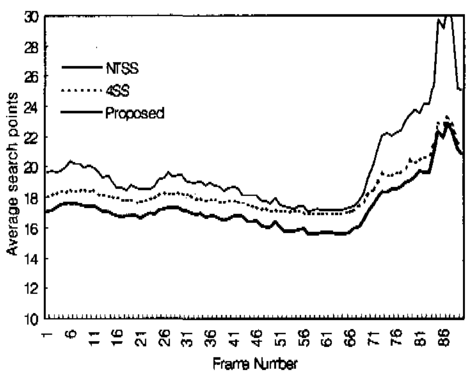
Table 2. Average Search Points per Motion Vector Estimation for 90 Frames

Search Algorithm	Claire	Salesman	Football	Tennis
FS	225	225	225	225
TSS	25	25	25	25
NTSS	18.85	18.00	19.91	22.58
4SS	17.85	17.30	18.44	20.10
ITSS	16.54	16.46	17.38	19.34

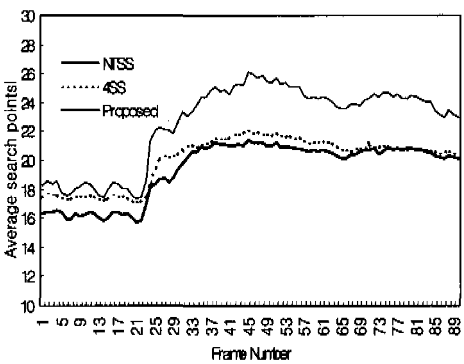
Table 3. The Probability to Find the True Motion Vector

Search Algorithm	Claire	Salesman	Football	Tennis
TSS	0.9430	0.9699	0.9294	0.6849
NTSS	0.9438	0.9925	0.9433	0.7940
4SS	0.9413	0.9737	0.9357	0.8048
ITSS	0.9432	0.9729	0.9534	0.8443

From the simulation results, we can see that the NTSS shows better MSE performance than other fast algorithms for the small-motion sequences. On the contrary, the 4SS yields good MSE performance for the fast-motion sequences. Nevertheless, the computational cost of both the NTSS and 4SS for the fast-motion sequences is relatively high. However, the proposed method maintains its performance better than the other fast algorithms in terms of both the MSE and computational requirement for all the sequences.



(a)



(b)

Fig. 2 Comparisons of NTSS, 4SS and ITSS on average search points per motion vector estimation versus frame number for (a)"Football" and (b)"Tennis" sequences.

Ⅳ. Conclusions

In this paper, we proposed an improved three-step search algorithm (ITSS) for block-based motion estimation. The ITSS using the center-biased search pattern can find small motion vectors effectively. In particular, we can reduce the computational complexity by using the pixel subsampling method for computing the MAD's at the search points in the first step. Experimental results indicated that the proposed method generally yields superior performance to some other popular fast search algorithms in terms of the computational requirement, motion compensation errors, and robustness.

References

[1] CCITT SGXV, "Description of reference model 8 (RM8)," Document 525, Working Party XV/4, Specialist Group on Coding for Visual Telephony, June 1989.

[2] ISO/IEC CD 11172-2 (MPEG-1 Video), "Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s: Video," 1993.

[3] ISO/IEC CD 13818-2 – ITU-T H.262 (MPEG-2 Video), "Information technology – Generic coding of moving pictures and associated audio information: Video," 1995.

[4] T. Koga, K. Ilinuma, A. Hirano, Y. Iijima, and Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Proc. NTC 81*, New Orleans, LA, Nov./Dec. 1981, pp. C9.6.1-C9.6.5

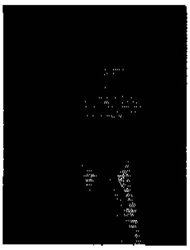
[5] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 148-157, Apr. 1993.

[6] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438-442, Aug. 1994.

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

[8] J. W. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 369-377, Aug. 1998.

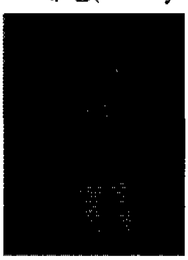
홍 원 기(Won-Gi Hong) 정회원



1985년 2월 : 고려대학교 전자공학과 졸업
1987년 2월 : 고려대학교 전자공학과 석사
1998년 8월 : 고려대학교 전자공학과 박사

1987년 1월~1993년 1월 : (주)삼성전자 S/W개발실 주임연구원
1993년 3월~1996년 8월 : 경인여자대학 전임강사
1998년 9월~현재 : 명지전문대학 전자과 조교수
<주관심 분야> 영상 부호화, 멀티미디어 통신

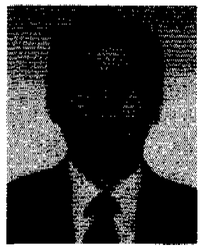
오 태 명(Tae-Myoung Oh) 정회원



1985년 2월 : 광운대학교 전자통신공학과 졸업
1997년 2월 : 고려대학교 전자공학과 석사
2000년 2월 : 고려대학교 전자공학과 박사수료

1986년 1월~1998년 3월 : (주)삼성 SDI 제품개발실 선임연구원
1998년 3월~현재 : 명지전문대학 전자과 조교수
<주관심 분야> 영상 부호화, 멀티미디어 통신

고 성 제(Sung-Jea Jo) 정회원



1980년 2월 : 고려대학교 전자공학과 졸업
1986년 8월 : State univ. of New York at Buffalo, 전기 및 컴퓨터공학과 박사

1988년 8월~1992년 5월 : The Univ. of Michigan Dearborn, 전기 및 컴퓨터공학과 조교수
1992년 3월~현재 : 고려대학교 전기전자전파공학부 교수
1997년~현재 : IEEE Senior Member.
1996년 11월 : IEEE APCCAS Best paper award.
1997년 : 대한전자공학회 해동논문상 수상
<주관심 분야> 신호 및 영상처리, 영상 압축 및 통신, 멀티미디어 통신등