論文
4-10-21

A Fast Multilevel Successive Elimination Algorithm
(빠른 다단계 연속 제거 알고리즘)

정수목(Soo-Mok Jung)*

ABSTRACT

In this paper, A Fast Multi-level Successive Elimination Algorithm (FMSEA) is presented for block matching motion estimation in video coding. Motion estimation accuracy of FMSEA is equal to that of Multilevel Successive Elimination Algorithm(MSEA). FMSEA can reduce the computations for motion estimation of MSEA by using partial distortion elimination technique. The efficiency of the proposed algorithm was verified by experimental results.

요 약

본 논문에서는 비디오코딩의 움직임 추정을 위한 빠른 다단계 연속제거 알고리즘(FMSEA: Fast Multilevel Successive Elimination Algorithm)을 제안하였다. FMSEA의 움직임 추정 정확도는 Multilevel Successive Elimination Algorithm(MSEA)과 동일하다. 부분 왜곡 제거 기법을 사용하는 FMSEA는 MSEA의 연산량을 효과적으로 줄일 수 있다. 실험을 통하여 제안된 알고리즘의 효율성을 확인하였다

Key words: Multilevel Successive Elimination Algorithm, Motion estimation, Motion vector

1. Introduction

Video frame has spatial redundancy and consecutive frames have very big temporal redundancy. The temporal redundancy can be removed by using motion estimation and compensation techniques. So, motion estimation and compensation techniques have been used widely in video coding.

Motion estimation methods are classified into two classes of block matching algorithms (BMA)[1][2] and pel-recursive algorithms (PRA)[3]. Owing to their implementation simplicity, block matching algorithms have been widely adopted by various video coding matching block of pixels within the search window in standards such as CCITT H.261[4], ITU-T H.263[5], and MPEG[6]. In BMA, the current image frame is partitioned into fixed-size rectangular blocks. The motion vector for each block is estimated by finding the best the previous frame according to matching criteria.

Although Full Search algorithm (FSA) finds the optimal motion vector by searching exhaustively for the

*정회원 삼육대학교 컴퓨터과학과 부교수

논문접수 : 2003. 10. 10.
심사완료 : 2003. 10. 16.

best matching block within the search window, its high computation cost limits its practical applications.

To reduce computation cost of FSA, many fast block matching algorithms such as three step search[7], 2-D log search[2], orthogonal search[8], cross search[9], one-dimensional full search[10], variation of three-step search [11][12], unrestricted center-biased diamond search[13] etc. have been developed. As described in[14], these algorithms rely on the assumption that the motion compensated residual error surface is a convex function of the displacement motion vectors, but this assumption is rarely true [15].

Therefore, the best match obtained by these fast algorithms is basically a local optimum.

Without this convexity assumption, Successive Elimination Algorithm (SEA) proposed by Li and Salari [16] reduces the computation cost of the FSA. To reduce the computation cost of SEA, Multilevel Successive Elimination Algorithm (MSEA) [17] were proposed. Our research team proposed Efficient Multilevel Successive Elimination Algorithms for Block Matching Motion Estimation (EMSEA) [18] to reduce the computation cost of MSEA. The partial distortion elimination scheme of EMSEA was improved and then

$$Bc(i,j)(m,n)=fc(i+m,j+n) \qquad (1)$$

$$\boldsymbol{B}_p^{(i,j,x,y)}(m,n)=f_p(i-x+m,j-y+n) \qquad (2)$$

where x and y represent two components of a candidate motion vector, $-M \le (x,y) \le M$ and $0 \le (m,n) \le N-1$. The SAD between the two blocks is defined as:

$$\mathrm{SAD}(x,y)=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\left|\boldsymbol{B}_c^{(i,j)}(m,n)-\boldsymbol{B}_p^{(i,j,x,y)}(m,n)\right| \qquad (3)$$

The goal of motion estimation is to find the best pair of indices (x,y) so that the following sum of absolute difference is minimized, as

The motion estimation accuracy of FMSEA is identical to that of FSA and the computation cost of MSEA is reduced by using FMSEA.

II. Multilevel Successive Elimination Algorithm

Let $f_c(i,j)$ and $f_p(i,j)$ denote the intensity of the pixel with coordinate (i,j) in the current frame and the previous frame respectively. Assume that the size of a block (Y component of the macro block in H.263) is NxN pixels, the search window size is $(2M+1)$x$(2M+1)$ pixels, and the matching criteria function is the sum of absolute difference(SAD) which is the distortion between two blocks.

Let $\boldsymbol{B}_c^{(i,j)}$ and $\boldsymbol{B}_p^{(i,j,x,y)}$ denote the target block and reference block in the current frame and the previous frame with the top left corners at (i,j) and $(i-x,j-y)$ respectively. $\boldsymbol{B}_c^{(i,j)}$ is the target block which requires motion vector.

follows:

$$d=\min_{x,y}\mathrm{SAD}(x,y) \qquad (4)$$

Applying mathematical inequality $|\,\|\boldsymbol{X}\|_1-\|\boldsymbol{Y}\|_1\,| \le \|\boldsymbol{X}-\boldsymbol{Y}\|_1$ [19] for $\boldsymbol{X}=\boldsymbol{B}_c^{(i,j)}$ and $\boldsymbol{Y}=\boldsymbol{B}_p^{(i,j,x,y)}$ gives

$$|R-M(x,y)| \le \mathrm{SAD}(x,y) \qquad (5)$$

where

$$R=\left\|\boldsymbol{B}_c^{(i,j)}\right\|_1=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\boldsymbol{B}_c^{(i,j)}(m,n)$$

$$M(x,y) = \left\| \boldsymbol{B}_p^{(i,j,x,y)} \right\|_1 \approx \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \boldsymbol{B}_p^{(i,j,x,y)}(m,n)$$

$$\mathrm{SAD}(x,y) = \left\| \boldsymbol{B}_c^{(i,j)} - \boldsymbol{B}_p^{(i,j,x,y)} \right\|_1 = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \left| \boldsymbol{B}_c^{(i,j)}(m,n) - \boldsymbol{B}_P^{(i,j,x,y)}(m,n) \right|$$

R and $M(x,y)$ are the sum norms of the two blocks and are pre-computed using the efficient procedure described in [16].

In MSEA, each block is partitioned into several sub-blocks. First, the block is partitioned into four sub-blocks with size $(N/2)$x$(N/2)$. Then each sub-block is partitioned into four sub-blocks with size $(N/4)$x$(N/4)$. This procedure can be repeated until the size of the sub-blocks become 2x2.

The maximum level of such partition is $L_{max}=\log 2^N-1$ for the blocks with size NxN. The MSEA with L-level partition, $0 \le L \le L_{max}$, is called L-level MSEA, and the SEA corresponds to the 0-level MSEA.

At the l-th level of the L-level MSEA, where $0 \le l \le L$, the number of the sub-blocks is $S_l = 2^{2l}$, and each sub-block is of the size N_lxN_l, where $N_l = N/2^l$. In L-level MSEA, we define $R_l^{(u,v)}$, $M_l^{(u,v)}(x,y)$, $\mathrm{SAD}_l^{(u,v)}(x,y)$, $\mathrm{SAD\_SB}_l(x,y)$ as follows:

$$R_l^{(u,v)} = \sum_{m=0}^{N_l-1}\sum_{n=0}^{N_l-1} \boldsymbol{B}_c^{(i,j)}(m+uN_l, n+vN_l) \qquad (6)$$

$$M(x,y) = \left\| \boldsymbol{B}_p^{(i,j,x,y)} \right\|_1 \approx \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \boldsymbol{B}_p^{(i,j,x,y)}(m,n)$$

(7)

$$\mathrm{SAD}(x,y) = \left\| \boldsymbol{B}_c^{(i,j)} - \boldsymbol{B}_p^{(i,j,x,y)} \right\|_1 = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \left| \boldsymbol{B}_c^{(i,j)}(m,n) - \boldsymbol{B}_P^{(i,j,x,y)}(m,n) \right|$$

(8)

$$\mathrm{SAD\_SB}_l(x,y) = \sum_{u=0}^{2^l-1}\sum_{v=0}^{2^l-1} \left| R_l^{(u,v)} - M_l^{(u,v)}(x,y) \right| \qquad (9)$$

where (u,v) is the index of a sub-block and $0 \le l \le L$. At l-th level partition, where $0 \le u,v \le 2^l - 1$, it can be easily seen that the following relation holds:

$$\mathrm{SAD}(x,y) = \sum_{u=0}^{2^l-1}\sum_{v=0}^{2^l-1} \mathrm{SAD}_l^{(u,v)}(x,y) \qquad (10)$$

$\mathrm{SAD\_SB}_l(x,y)$ can be classified as equation (11).

In [17], X. Q. Gao etc. proved that equation (5) can be expressed as equation (12) and then they proposed the MSEA which is based on equation (12).

$$\mathrm{SAD\_SB}_l(x,y) = \begin{cases} |R - M(x,y)| & l = 0 \\ \sum_{u=0}^{2^l-1}\sum_{v=0}^{2^l-1} \left| R_l^{(u,v)} - M_l^{(u,v)}(x,y) \right| & 1 \le l \le L \\ \mathrm{SAD}(x,y) & l = L+1 \end{cases}$$

(11)

$$\mathrm{SAD\_SB}_l(x,y) \le \mathrm{SAD\_SB}_{l+1}(x,y) \qquad (12)$$

where $0 \le l \le L$.

The L-level MSEA procedure is as follows:

1 select an initial candidate motion vector within the search window in the previous frame.

2 calculate the motion vector (***MV***) and the SAD at the selected search point. These ***MV*** and SAD become the current temporary motion vector and the current minimum SAD respectively.
(curr_temp_***MV***=***MV***,
curr_min_SAD= SAD)

3 select another search point among the rest of the search points.

4.0 calculate the $\mathrm{SAD\_SB}_0$ at the selected search point. if(curr_min_SAD$\le$SAD_SB$_0$) then goto 7.

4.1 calculate the $\mathrm{SAD\_SB}_1$ at the selected

search point.
if(curr_min_SAD≤SAD_SB1)
then goto 7.
.
.
.

4.L calculate the SAD_SB$_L$ at the selected search point.
if(curr_min_SAD≤SAD_SB$_L$)
then goto 7.

5 calculate the SAD at the selected search point.
if (curr_min_SAD≤SAD) then goto 7.

6 curr_min_SAD=SAD
calculate the motion vector at the selected point. This motion vector becomes the current temporary motion vector.(curr_temp_***MV***=***MV***)

7 if (all the search points in the search window are not tested?) then goto 3.

8 the optimum motion vector= curr_temp_***MV***,
the global minimum SAD= curr_min_SAD.

The MSEA speeds up the process of finding the motion vector by eliminating hierarchically impossible candidate motion vectors in the search window before their SAD calculation which requires intensive computations.

If we delete step 4.0~4.L in the L-level MSEA procedure, then the procedure becomes the full search algorithm procedure. If we delete step 4.1~4.L, then the procedure becomes the successive elimination algorithm procedure.

III. Fast Multilevel Successive Elimination Algorithm

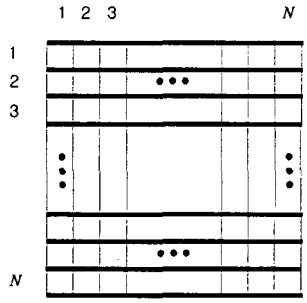
As shown MSEA procedure step 4.1~~ ‖ ~~4.L, MS」 EA can eliminate impossible candidate vectors before their SADs are computed. So, MSEA can reduce the computations of SEA.

If Partial distortion elimination (PDE)[20] technique is applied to MSEA, then the computations of SADs calculation can be reduced. In MSEA, SADs calculation must actually be computed for all NxN pixels.

PDE is an effective speed up technique used in vector quantization to find a best reconstruction vector from a set of vector code words.

All of the terms in the equation (3) are positive. If at any point the partially evaluated sum exceeds the current minimum SAD($curr\_SAD_{min}$), that candidate block cannot be the best matching block and the remainder of the sum does not need to be calculated. While it is not efficient to test the partial sum against the $curr\_SAD_{min}$ every additional term is added, a reasonable compromise is to perform the test after N times additions when block size is NxN as shown in fig2. So, the maximum number of PDE test is N times.

In MSEA, SAD is calculated all the NxN pixels and then the calculated SAD is compared with $curr\_SAD_{min}$.



[Fig. 2] PDE test in SAD calculation

If step 5 of MSEA is replaced with the following steps, then MSEA procedure becomes that of FMSEA.

step 5.0 calculate the SAD at the

selected search block for N pixels.

step 5.1 if ($curr\_SAD_{min} \leq SAD$) then goto step 7.
else if SAD is not calculated for all pixels(NxN) then goto step 5.0

IV. Experimental Results of Fast Block Sum Pyramid Algorithm

The standard video sequences with qcif(176x144) such as "foreman.qcif", "trevor.qcif" were used in the experiment and we tested 100 frames of the sequences. The block size was 16x16 pixels (N=16). The size of the search window was 31x31 pixels (M=15) and only integer values for the motion vectors were considered. Experimental results are shown in table 1. In table 1, "m.e." means matching evaluation that require SAD calculation, avg. # of rows means the number of calculated rows in the SAD calculation before partial distortion elimination. Overhead(in rows) is the sum of all the computations except SAD calculation. "in rows" means that the computations are represented in order of 1 row SAD computations.

It is important to notice that with the MSEA, the efficiency of the procedure depends on the order in which the candidate motion vectors are searched, and that the most promising candidates should be tested first. This eliminates the maximum number of candidates. In our experiment, we used spiral search.

Table 1. The computations of FMSEA

Algo-rithm	Test seq.	Avg. # of m.e./frame	Avg. # of rows/m.e.	Overhead (in rows)	Total (in rows)	re-duction
MSEA (0)	foreman	19,283.6	16.00	8,657.5	317,195	
	trevor	19,497.2	16.00	8,669.7	320,625	
FMSEA (0)	foreman	19,283.6	6.01	8,659.2	124,554	60.7%
	trevor	19,497.2	7.28	8,670.1	150,610	53.0%
MSEA (1)	foreman	4,804.9	16.00	13,463.3	90,342	
	trevor	7,879.3	16.00	13,511.6	139,580	
FMSEA (1)	foreman	4,804.9	9.72	13,461.0	60,165	33.4%
	trevor	7,879.3	10.85	13,510.5	99,001	29.1%
MSEA (2)	foreman	1,8000.5	16.00	18,161.0	46,969	
	trevor	4,290.1	16.00	21,288.6	89,930	
FMSEA (2)	foreman	1,8000.5	12.71	18,159.3	41,044	12.6%
	trevor	4,290.1	12.29	21,290.9	74,016	17.7%
MSEA (3)	foreman	749.0	16.00	24,975.2	36,959	
	trevor	1,563.4	16.00	38,071.1	63,086	
FMSEA (3)	foreman	749.0	14.63	24,971.4	35,929	2.8%
	trevor	1,563.4	13.70	38,069.7	59,488	5.7%

The FMSEA which incorporates the MSEA with PDE, reduces the computations of MSEA by 60.7%, 53.0% maximally for foreman.qcif, trevor.qcif respectively.

V. Conclusions

A Fast Mutilevel Successive Elimination Algorithm based on MSEA has been proposed to reduce the computations of MSEA for motion estimation in video coding. Partial distortion elimination technique used in EMSEA was improved and then the improved PDE technique was applied to MSEA. The FMSEA can find the global optimum solution and outperforms the MSEA and can reduce the computations of block matching calculation of MSEA. FMSEA is a very efficient solution for video coding applications that require both very low bit-rates and good coding quality.

References

[1] H. G. Musmann, P. Pirsh, and H. J. Gilbert, "Advances in picture coding," Proc. IEEE, vol. 73, pp. 523-548, Apr. 1985.

[2] J. R. Jain, and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol. COMM-29, pp. 1799-1808, Dec. 1981.

[3] A. N. Netravali, and J. D. Robbins, "Motion compensated television coding: Part I," Bell Syst. Tech. J, vol. 58, pp. 631-670, Mar. 1979.

[4] CCITT Standard H.261, "Video codec for audiovisual services at px64 kbit/s," ITU, 1990.

[5] ITU-T DRAFT Standard H.263, "Video coding for narrow telecommunication channel at (below) 64kbit/s," ITU, Apr. 1995.

[6] ISO-IEC JTC1/SC2/WG11, "Preliminary text for MPEG video coding standard," ISO, Aug. 1990.

[7] T. Koga, K. Iinuma, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," Proc. Nat. Telecommunications Conf., pp. G5.3.1- G.5.3.5, Nov. 1981.

[8] A. Puri, H.-M. Hang, and D. L. Schilling, "An efficient block matching algorithm for motion compensated coding," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 25.4.1-25.4.4, 1987.

[9] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans. Commun., vol. 38, no. 7, pp. 950-953, July 1990.

[10] M. J. Chen, L. G. Chen and T. D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 504-509, Oct. 1994.

[11] H. M. Jong, L. G. Chen, and T. D. Chiueh, "Accuracy improvement and cost reduction of 3-step search block matching algorithm for video coding," IEEE Trans. Circuits Syst. Video Technol, vol. 4, pp. 88-91, Feb. 1994.

[12] 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.

[13] J. Y. Tham, S. Ranganath, M. Ranganath, and A. Ali Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 8, no. 4, pp. 369-377, Aug. 1998.

[14] B. Liu, and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," IEEE Trans. Circuits Syst. Video Technol., vol. 3, no. 2, pp. 148-157, Apr. 1993.

[15] K. H. K. Chow, and M. L. Liou, "Genetic motion search algorithm for video compression," IEEE Trans. Circuits Syst. Video Technol., vol. 3, pp. 440-446, Dec. 1993.

[16] W. Li, and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Trans. Image Processing, vol. 4, No.1, pp. 105-107, Jan. 1995.

[17] X. Q. Gao, C.J. Duanmu, and C.R. Zou, "A Multilevel Successive Elimination Algorithm for Block Matching Motion Estimation," IEEE Trans. Image Processing, Vol. 9, No.3, pp.501-504, Mar. 2000.

[18] S. M. Jung, S. C. Shin, H. Baik, and M. S. Park, "Efficient Multilevel Successive Elimination Algorithms for Block Matching Motion Estimation," IEE Vision and Image Signal Processing, vol.149, No.2, pp. 73-84, Apr. 2002.

[19] T. M. Apostol.: Mathematical Analysis, Reading, MA: Addison- Wesley, 1975.

[20] Y. Q. Zhang and S. Zafar, "Predictive Block Matching Motion Estimation for TV Coding-part II: Interframe Prediction," IEEE Trans. Broadcast., vol. 37, pp. 102- 105, Sept. 1991.

정 수 목

1984년 2월 : 경북대학교 전자공학과졸업(공학사)
1986년 2월 : 경북대학교 대학원전자공학과졸업 (공학석사)
1986년 1월 ~ 1991년 2월 LG정보통신 연구소 연구원
2002년 2월: 고려대학교 대학원 컴퓨터 학과 졸업(공학박사)
1988년 3월~현재 : 삼육대학교 컴퓨터학과 부교수
관심분야 : 멀티미디어, 영상처리, 컴퓨터보안