Multi-Level Motion Estimation Algorithm Using Motion Information in Blocks

Heak-Bong Kwon†

ABSTRACT

In this paper, we propose a multi-level block matching algorithm using motion information in blocks. In the proposed algorithm, the block-level is decided by the motion degree in the block before motion searching procedure, and then adequate motion searching performs according to the block-level. This improves computational efficiency by eliminating the unnecessary searching process in no motion or low motion regions, and brings more accurate estimation results by deepening motion searching process in high motion regions. Simulation results show that the proposed algorithm brings the lower estimation error about 20% MSE reduction with the fewer blocks per frame and the operation number was reduced to 56% compared to TSSA and 98% compared to FS-BMA with constant block size.

블록 내의 움직임 정보를 이용한 다단계 움직임 예측 알고리즘

권 혁 봉†

요 약

본 논문에서는 블록 내의 움직임 정도에 따라 다단계의 블록 레벨을 설정하고, 각 블록 레벨에 따라 각기 다른 움직임 예측을 수행하는 다단계 움직임 예측 알고리즘을 제안한다. 이는 움직임이 없거나 적은 영역에서의 잉여 탐색 과정을 제거하여 계산적 효율성을 증대하고, 움직임 정도가 큰 영역에 대해서는 움직임 예측 과정을 심화시켜 예측의 정확성을 향상시킨다. 실험결과 프레임 당 적은 수의 블록으로 고정된 크기의 블록을 가진 전역 탐색 블록 정합 알고리즘과 Three-Step 탐색 알고리즘보다 MSE를 20% 정도 감소시켰으며, 연산량은 전역 탐색 블록 정합 알고리즘과 비교하여 98%, Three-Step 탐색 알고리즘과 비교하여 56% 정도 절감하는 효과를 얻을 수 있었다.

Key words: motion estimation, FS-BMA, TSSA, VS-BMA, video coding

1. Introduction

Recently, the development of computer technology and high speed network environment are gradually extending the range of multimedia communication. The studies on the standardization of multimedia data and compression technologies according to this trend have actively progressed and one of them is video coding. The video coding uses the characteristic of high temporal correlation in each frame of the image sequence. Therefore, the motion estimation that eliminates the temporal redundancy in each frame is the core part of the video coding. There are two representative algorithms of motion estimation: block matching algorithm(BMA)[1] and pel-recursive algorithm(PRA)

이 논문은 2003학년도 김포대학 연구비 지원에 의하여 연구되었음.
접수일 : 2002년 11월19일, 완료일 : 2003년 1월 8일
† 정회원, 김포대학 전자정보계열 조교수

[2]. The BMA is generally used and suitable for the real time processing since its computational complexity is much lower than the PRA. It has already been adopted by all the international video coding standards like H.261/263 and MPEG-1/2.

The BMA estimates the motion by using the correlation between successive frames of an image sequence. First, the BMA partitions an image into a set of non-overlapped, equally spaced, fixed size, small rectangular blocks, then finds the best matched block in the search area of previous frame. Full search block matching algorithm(FS-BMA) is the representative method of all block matching algorithms. The FS-BMA, that thoroughly examines all locations in a candidate search area, provides an optimal performance. However, the FS-BMA requires too much operational loads compared with its performance, which has motivated the development of the fast block matching algorithms such as a three-step search algorithm(TSSA)[3] and a cross search algorithm that increase searching speed[4]. For the purpose of image quality improvement, variable size block matching algorithm(VS-BMA)[5] and multi-level block matching algorithm(ML-BMA)[6] had been researched, too. Compared to the FS-BMA, VS-BMA is superior to the local motion estimation and this algorithm causes fewer motion estimation errors but additional computational loads can occur.

In this paper, we propose a multi-level motion estimation algorithm that performs different motion estimation according to each block level, respectively. In result, the proposed algorithm improves computational efficiency by eliminating the unnecessary searching processes in low motion block and also increases the exactness of estimation by the deepening motion estimation procedure in high motion block.

2. Multi-level motion estimation algor

The FS-BMA and TSSA perform the motion estimation with the fixed size blocks. These kinds of motion estimation method are based on the assumption that all the pixels in the blocks have the same motion from frame to frame. Therefore, each motion cannot be estimated separately when there are two or more different motions in a block, resulting in a lower accuracy of estimation. Also, the FS-BMA examines all the search areas while moving the correlation window, which causes computational loads. In the FS-BMA, if the maximum displacement that the block can move is w, the size of the search area for the previous frame is w, the size of the search area for the previous frame is $(N+2w)\times(N+2w)$, and the number of search points that passes by during the search process of a block is $(2w+1)^2$. Under the same conditions, the number of search points for the TSSA is $1+8[\log_2(w+1)]$, which can reduce the computational loads of the FS-BMA, but the motion estimation error can relatively increase or the trapping to a local minimum point can occur, resulting in the deterioration of image quality. In addition, the three step search process is performed even there are no motions between the previous and present frames, and for the low motion blocks, which lowers the computation efficiency.

As seen above, applying the constant estimation method to all blocks is inefficient in estimation accuracy and matching speed[7]. Therefore, to make an accurate estimation in a short time, an adaptive estimation method according to the motion information in the block is required.

In this paper, we propose a multi-level motion estimation algorithm using the motion information in the block to improve the block matching speed and image quality. Three block levels will be set according to the motion degree in the block, and the block size will be varied according to the motion degree in each block level. Also, different search steps and ranges will be applied to the motion estimation of each block level. Fig. 1 shows the flow chart of the proposed algorithm and search process at each level.

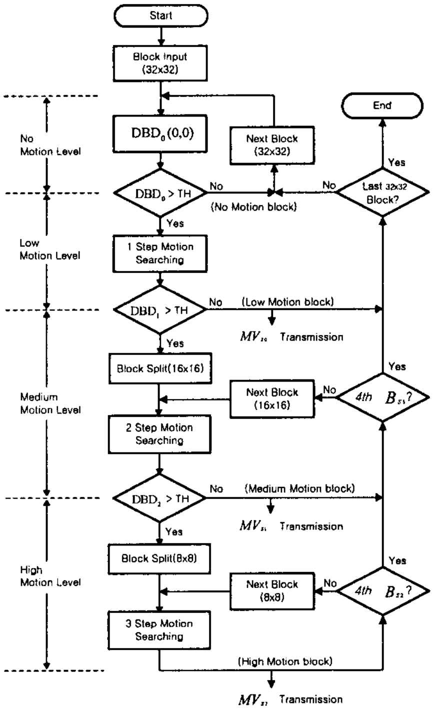


Fig. 1. The flow chart of proposed algorithm

2.1 Setting of the multi-level block

To set the multi-level block, first of all, the motion of block must be classified into one of the four kinds of motion levels. The blocks that have motions will be set each three-levels for degree of motion beginning from the low motion block level to the high motion block level and each levels will be set into a different sized block. This is for the accurate estimation of the local motions in the block.

The DBD(Displaced Block Difference) between two continuous frames means the similarity so its value can be found using Equation (1).

$$DBD\,(x,y)=\frac{1}{N\times N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|F_n(i,j)-F_{n-1}(i+x,j+y)|$$
$$x,y\in[-w,-w+1,\ldots,w-1,w] \qquad (1)$$

Where, N is the width and the height of the block, F_n is the block to be predicted within the present frame, and F_{n-1} is the candidate block within the search area of the previous frame.

The block partition process is performed by comparing the value of DBD found from each block level with the threshold using Equation (2).

$$\begin{cases} if\ \ DBD_n(i,j)\geq TH_{no\text{-}motion}, & Split \\ otherwise, & Terminated \end{cases} \qquad (2)$$

Fig. 2 shows the multi-level block setting process according to the degree of motion. First, the DBD_0 (0,0) is computed for the 32×32 size block and then goes through no motion block level using Equation (2) to check if there is motion[8]. The 32×32 size blocks, which have been discriminated as having motions when their value of $DBD_0(0,0)$ has exceeded the threshold in the no motion block level, go through the low motion block level. In this level, through searching, which is the first step of the three step search process, the minimum displacement is found. And then the value of DBD_1 is calculated and checked if the value of DBD_1 is bigger than the threshold. The blocks that have exceeded the threshold in the low motion block level are divided into the 16×16 size blocks in the medium motion block level. The minimum displacement is found through the two steps search on the 16×16 size blocks, then DBD_2 is calculated and checked if its value is bigger than the threshold. For the blocks that exceed the threshold in the medium motion block level, they are divided again into the 8×8 blocks and go through the high motion block level in which all of the three step search are performed, and then the processing of one block is finished.

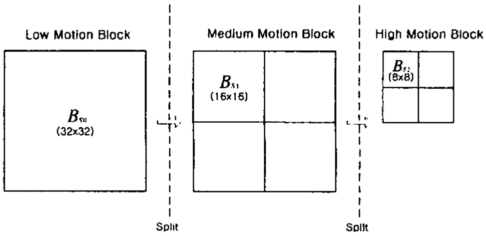


Fig. 2. The setting process of multi-level block

2.2 Multi-step motion estimation

In the multi-level motion estimation, different search steps and displacements of search area are applied to each block level according to the motion degree in the block. It intends to increase the efficiency of the search process by performing the motion estimation process adaptively according to the degree of motion in the block. First, the motion estimation process is omitted in the no motion block level. In the low motion block level, one step search is performed as shown in Fig. 3, because the degree of motion in the block is small, and the displacements of search area is $(\pm1,\pm1)$. The size$(\Delta x,\Delta y)$ of the motion vector(MV_{s0}) is $(\pm1,\pm1)$. In the medium motion block level, the minimum displaced point found in the low motion block level is set as the center point and only two steps of searching process are performed, and the displacement of search area is $(\pm3,\pm3)$. The size$(\Delta x,\Delta y)$ of the motion vector (MV_{s1}) is $(\pm3,\pm3)$. In the high motion block level, the TSSA is applied to the motion estimation of a block with a large motion, and the displacement of search area is $(\pm7,\pm7)$. The size of the motion vector(MV_{s2}) obtained from this level is $(\pm7,\pm7)$. Therefore, the final motion vector is obtained by adding the motion vector of each level, as shown in Equation (3).

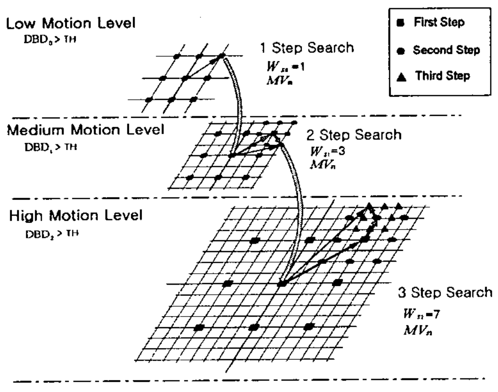


Fig. 3. The procedure of multi-level motion estimation.

$$MV_{final} = MV_{S0} + MV_{S1} + MV_{S2} \tag{3}$$

The motion vectors obtained from each block level are inherited to the lower block level, so if the motion estimation of whole three levels is performed then a maximum size $(\pm11,\pm11)$ of motion vector can be obtained by the sum of the motion vectors from all block levels.

2.3 Setting the search area

The motion information of the moving object varies smoothly and slowly, and the moving object usually exists over the blocks so the motion vectors have a high correlation among the adjacent blocks. Therefore, by setting the position of the search area through the use of the motion vectors of adjacent blocks, the matching feasibility can be increased.

In this paper, the motion vectors of adjacent blocks that were already obtained as shown in Fig. 4 are used for the setting of the search area. First, the DBD at B(i, j) is calculated, and four DBDs are calculated by each motion vector of four adjacent search blocks. Then, the search area is decided by the minimum vector between DBD at B(i, j) and other four DBDs of adjacent blocks. This is the case for the highest level block B_{s0}, and for the low level blocks(B_{s1},B_{s2}) the motion vector acquired in the upper level is used. Therefore, the inheritance of the motion vector from the upper level allows the acquisition of a large motion vector though the search is set narrowly.

B(i-1,j-1)	B(i,j-1)	B(i+1,j-1)
B(i-1,j)	B(i,j)	

Fig. 4. Setting the search area using motion vectors in adjacent blocks.

3. Results of experiments and consideration

To evaluate the performance of the proposed multi-level motion estimation algorithm, 163 frames (1 ~ 163) of the Claire and the Salesman image (CIF: 352×288) sequences were used. Each frame has a size of 352×288(CIF) and the motion estimation was performed every three frames. In the proposed algorithm, the initial block size is 32×32, and it is divided up to the 16×16 and 8×8 size according to each block level. The size of the displacement w increases from $W_{s0}=1$ to $W_{s1}=3$ and $W_{s2}=7$ as the block level increases, and through all three levels its size is $W_{Total}=11$. Fig. 5 shows the block formation process of the Claire and Salesman image according to the amount of motion in the blocks. The region where the block doesn't set means it is discriminated as a region of no-motion. The blocks with motions show that they can be divided up to the 16×16 and 8×8 size according to the motion degree from the initial block size of 32×32.

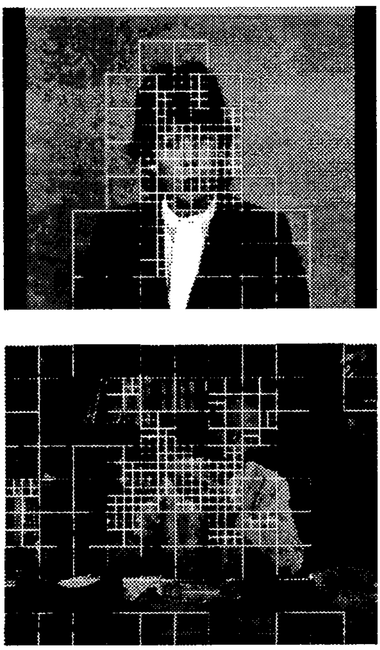


Fig. 5. Block formation according to motion degree in blocks of Claire and Salesman Images.

The comparative algorithms are the FS-BMA, VS-BMA and TSSA. The block size for the FS-BMA and TSSA is set to 16×16. For the VS-BMA, the size of the initial block is set to 32×32 and it varies according to the degree of the motion as the proposed algorithm does. The displacement of area (w) was set to W=11 for the FS-BMA, VS-BMA and the proposed algorithm, and for the TSSA set to W=7 which is the maximum allowable size[6]. Also, the computational loads and the MSE(Mean Square Error) have been compared to the FS-BMA and TSSA in the case of the block size is 8×8, which is the minimum block size of the proposed algorithm. For the comparison of the performances, the MSE of Equation (4) was used.

$$MSE = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left[F_n(i,j) - F_{n-1}(i,j) \right]^2 \tag{4}$$

Table 1. shows the performance comparison between the comparative algorithms and proposed algorithm. In the Table 1, the MSE is the value of one pixel, and the relative number of operations shows the number of operations for each algorithm when the operation number of the FS-BMA in the 16×16 size is assumed to be 1. The computational complexity of each algorithms was compared by calculating the number of operations which are used in the process of the motion estimation[9]. In the case of FS-BMA, the block size is 16×16 and the displacement of the search area is set to W=11, then the number of points to search is $(2W+1)^2$=529. The motion vector required 529×(256×3+1) operations for the 16×16 size block. In conclusion, the operation number for the estimation of one frame in the CIF image (396 block) is 529×(256×3+1)×396= 161,093,196.

In the Claire image, the proposed algorithm performs motion estimation with about 34% smaller number of blocks than the FS-BMA and TSSA of the 16×16 size, and yet has a lower value of MSE about 20%. The operation number reduced to 59%

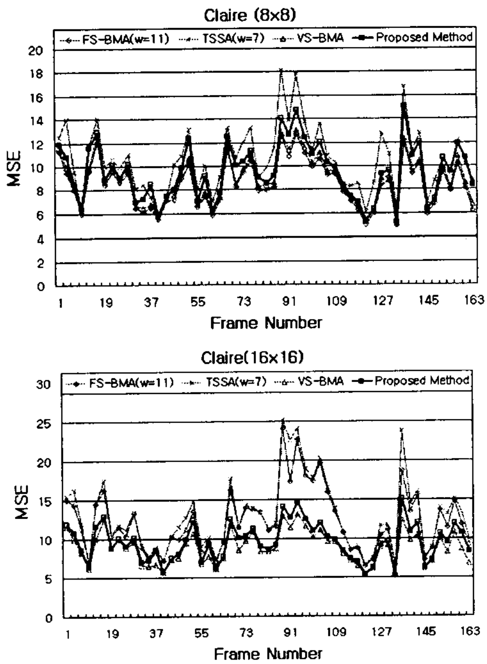
Table 1. Performance comparison between the comparative algorithms and the proposed algorithm

	Algorithm	MSE/pixel	Number of relative operations	Number of blocks/Frame
Claire (CIF)	FS-BMA 8×8 (w=11)	8.4996	0.9987	1584
	TSSA 8×8 (w=7)	10.5875	0.0474	1584
	FS-BMA 16×16 (w=11)	12.1907	1	396
	TSSA 16×16 (w=7)	12.9197	0.0473	396
	VS-BMA	8.8464	1.3559	339.8
	Proposed Method	9.56919	0.0195	264.2
Salesman (CIF)	FS-BMA 8×8 (w=11)	22.0192	0.9987	1584
	TSSA 8×8 (w=7)	27.3110	0.0474	1584
	FS-BMA 16×16 (w=11)	32.9167	1	396
	TSSA 16×16 (w=7)	36.6463	0.0472	396
	VS-BMA	24.0376	1.4224	368.3
	Proposed Method	27.9298	0.0229	301.1

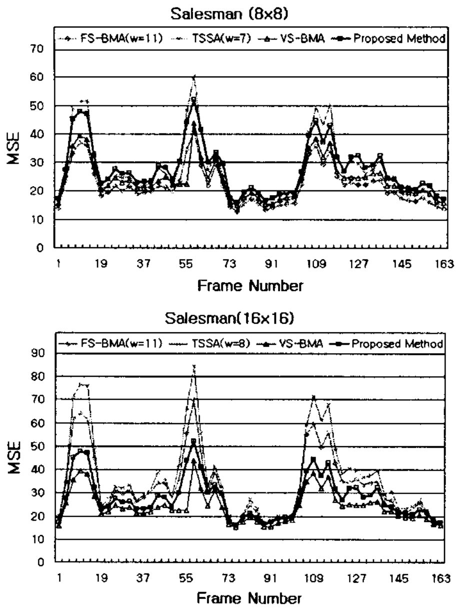
compared to TSSA and 98% compared to FS-BMA. Also, about 10% lower MSE was obtained than TSSA of the 8×8 size, and the value of the MSE had only a difference of about 12% with 98% fewer number of operations than the FS-BMA. Therefore, the proposed algorithm reduces the operation number and the motion vectors to a considerable amount while keeping the image quality close to the FS-BMA level. In the VS-BMA, it has the lowest value of the MSE, but its computational load is terrible. The average number of created blocks for each level was about 22.2 for B_{s0}, 80.9 for B_{s1}, and 241.9 for B_{s2} in the Claire image. The threshold was set by selecting the value with the minimum MSE within the range where the number of created blocks is similar to the number of blocks of the comparative algorithm in the 16×16 size.

Likewise in the Salesman image, compared with FS BMA and TSSA of the 16×16 size, the value of MSE decreased to about 20% with about 24% smaller number of blocks respectively. The operation number reduced to about 52% compared to TSSA, and about 98% compared to FS-BMA. Compared with the 8×8 size TSSA the MSE was similar, and compared to the 8×8 size FS-BMA it showed a 27% difference. It is caused by the more complicate image of Salesman than Claire image, and from Table 1 it can be seen that the value of MSE of all algorithms is increased. In the experiment of the Salesman image sequences, the average number of blocks created in each level was about 33.2 for B_{s0}, about 116.8 for B_{s1}, and 294.3 for B_{s2}. Fig. 6 shows the MSE values of the proposed algorithm and the comparative algorithms at a 3-frame interval.

The MSE values of each algorithm have been compared when the block size is 8×8 and 16×16. In almost all frames, the performance of the proposed algorithm is superior to the comparative algorithms of the 16×16 size, and it also can be seen that the proposed algorithm outperforms especially when the



(a) Claire



(b) Salesman

Fig. 6. MSE ratio on the images of Claire and Salesman.

degree of motion between the frames is large which decreases the performance of the comparative algorithms.

4. Conclusion

In this paper, we proposed an algorithm for efficient motion estimation using the motion information in the blocks. While the comparative algorithms were not adaptive to the regional motions in the block, the proposed algorithm varied the size of blocks according to the degree of motions, and applied the multi-block level that divided the block more and more as the motion got bigger and bigger. To do this, the redundant motion search process is removed for the blocks that have no motions, and the size of the search block is varied according to the degree of the motion for the block have motions within. Even in the case of the motion estimation error occurred by the getting trapped in the local minimum point during the high block level searching process, the method to vary block size will correct such errors in the lower block level. It supplements the problem of image quality deterioration caused by the convergence to the local minimum point in the TSSA. Also, a multi level motion estimation process was set up in each block where the searching level and range was applied differently, reducing the computational load and time required for searching.

The experiment results showed that the proposed algorithm had kept the image quality while reducing the operation number and the motion vector number considerably compared to the FS-BMA and TSSA of the 8×8 size. Also, in comparison with FS-BMA and TSSA of the 16×16 size, which has the similar number of blocks with the proposed algorithm, the proposed algorithm reduced about 21% MSE more than FS-BMA and about 26% more than TSSA. It reduced the operation number to 59% of TSSA and 98% of FS-BMA, allowing fast block matching.

Consequently, the proposed multi-level motion estimation algorithm using the motion information in the blocks is proved through experiments that it improves the image quality more than the FS-BMA and TSSA while allowing fast block matching. In addition, if the threshold is optimized according to the motion characteristics within the image, the advanced performance can be achieved, and more research will have to follow.

References

[1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Communication,* vol. COM 29, no. 12, pp. 1799-1808, December 1981.

[2] C.B. Bergeron, E. Dubois, "Gradient Based Algorithm for Block Oriented MAP Estimation of Motion and Application to Motion-Compensated Temporal Interpolation," *IEEE Trans. On Circuits and Systems for Video Technology,* vol.1, no.1, Mar. 1991.

[3] T. Koga, K. Linuma, A. Hirano, Y. Ujima, and

T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *Proc. NTC'81, New Orleans, LA*, Nov. 1981, pp. G5.3.1-G5.3.5.

[4] H. Gharavi, and M. Mills, "Block matching motion estimation algorithms-new results," *IEEE Trans. Circuits Syst.*, pp. 649-651, 1990.

[5] A. Puri, H. M. Hang, and D. L. Schilling, "Interframe coding with variable block size motion compensation," *Proc. GLOBECOM'87, Tokyo, Japan*, pp. 2.7.1-2.7.5, Nov. 1987.

[6] D. S. Shin, N. J. Kwak, H. B. Kwon and J. H. Ahn, "Multi Level, Multi Step Motion Estimation Algorithm," *IEICE Trans. Information and Systems*, vol.E84 D no.6, pp. 760-762, 2001.

[7] D. S. Shin and J. H. Ahn, "Fast Variable-size Block Matching Algorithm for Motion Estimation Based on Bit-pattern," *Journal Of Korea Multimedia Society*, vol. 3 no.4, pp. 372-379, 2000.

[8] H. S. Oh, C. H. Lee, H. K. Lee, and J. H. Jeon, "A new block matching algorithm based on an adaptive search area adjustment using spatio-temporal correlation," *IEEE Trans. Consumer Electronics*, vol. 45 no. 3, pp. 745-752, 1999.

[9] J. Chalidabhongse, and J.Kuo, "Fast motion vector estimation using multiresolution spatio-temporal correlations," *IEEE Trans. Circuits and Systems for Vidieo Technology*, vol. 7 no.3, pp. 477-488, 1997.

권 혁 봉

1889년 2월 호서대학교 정보통신공학과(학사)
1992년 2월 호서대학교 정보통신공학과(석사)
2001년 8월 충북대학교 정보통신공학과(박사)
1997년~현재 김포대학 전자정보계열 조교수

관심분야 : 영상통신 및 영상정보처리, 컴퓨터비전, 신호 및 시스템

E-mail : hbkwon@kimpo.ac.kr

교신저자

권 혁 봉 415-873 경기도 김포시 월곶면 포내리 산 14-1 김포대학 전자정보계열