Fast Variable-size Block Matching Algorithm for Motion Estimation Based on Bit-patterns

비트패턴 기반 움직임 추정을 위한 고속의 가변 블록 정합 알고리즘

권혁봉
김포대학 전자정보계열

송영준
충북대학교 정보통신공학과

Heak-Bong Kwon (hbkwon@kimpo.ac.kr)
Division of Electronic and Information Science, Kimpo College
Young-Jun Song (songyjorg@dreamwiz.com)
Dept. of Computer and Communication Engineering, Chungbuk University

중심어 : 움직임 예측, 블록 매칭, FS-BMA

Keyword : Motion estimation, Block matching, FS-BMA

요 약

본 논문에서는 비트패턴을 기반으로 한 고속의 가변 블록 움직임 예측 알고리즘을 제안한다. 제안된 방법은 블록 내의 평균값을 기준으로 8bit 화소값을 0과 1의 비트패턴으로 변환한 후 블록의 움직임 예측을 수행한다. 비트변환을 통한 영상의 단순화는 움직임 추정의 계산적 부담을 감소시켜 빠른 탐색을 가능하게 한다.

그리고 블록 내의 움직임 정도를 미리 판별하여 이를 기반으로 한 적응적 탐색이 불필요한 탐색을 제거하고 움직임이 큰 블록에서는 정합(matching) 과정을 심화시켜 보다 빠르고 정확한 움직임 예측을 수행한다.

본 제안된 방식을 가지고 실험한 결과, 한 프레임(frame) 당 적은 수의 블록으로 고정된 크기의 블록을 가진 전역 탐색 블록 정합 알고리즘(full search block matching algorithm; FS-BMA)보다 예측 에러를 적게 발생시켜 평균적으로 0.5dB 정도의 PSNR 개선을 가져왔다.

Abstract

In this paper, we propose a fast variable-size block matching algorithm for motion estimation based on bit-patterns. Motion estimation in the proposed algorithm is performed after the representation of image sequence is transformed 8-bit pixel values into 1-bit ones by the mean pixel value of search block, which brings a short searching time by reducing the computational complexity.

Moreover, adaptive searching methods according to the motion information of the block make the procedure of motion estimation efficient by eliminating unnecessary searching processes of low motion block and deepening a searching procedure in high motion block.

Experimental results show that the proposed algorithm provides better performance - average 0.5dB PSNR improvement and about 99% savings in the number of operations - than full search block matching algorithm with a fixed block size.

I. Introduction

The advancement in computer technology and the high-speed network environment of today enable multimedia data to be transmitted widely. However, the characteristics of multimedia data that require large bandwidth and storage media and the demands of users for real-time processing have not been fulfilled sufficiently in the current communication system. For this reason, the standardization of multimedia data and the studies of data compression have been proceeding actively. One of those is video coding, which aims to provide high quality moving images at low cost. Video compression exploits the high

접수번호 : #030602-001
접수일자 : 2003년 6월 2일, 심사완료일 : 2003년 6월 17일

temporal correlation between successive frames of an image sequence. Therefore, the motion estimation that eliminates the temporal redundancies between frames is the core part in video compression. The simple and effective way of the block matching algorithm (BMA)[1] is mainly used for the motion estimation. This algorithm has already been adopted in various video coding standards like H.261/263 and MPEG. The block matching algorithm allocates one motion vector (MV) per one block on the assumption that all pixels in one block have uniform motion. As the size of the block becomes larger, the higher compression rates can be obtained due to the transmission of a lower number of motion vectors, but image quality is rather inferior because of many prediction errors that may arise.

Block matching is the process that finds a candidate block, within a search area in the previous frame, which is most similar to the search block in the present frame. Full search block-matching algorithm (FS-BMA) that exhaustively examines all locations in a search area provides an optimal solution. However, its computational loading has motivated the development of block matching algorithms such as the three-step search[2] and the cross-search algorithm[3]. Though these algorithms are inferior in prediction accuracy, they can improve searching speed. On the other hand, The variable block size model for improving the image quality problem has been researched[4], which is the reason that the regional motion change of the image can't be reflected effectively in the case of the block size being fixed. The variable block size method causes fewer prediction errors than those of a full search block-matching algorithm; additional computational loading can occur.

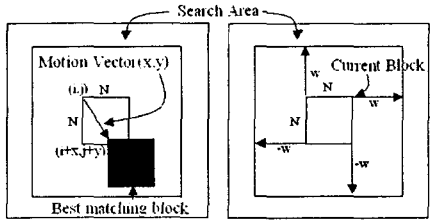
The proposed block matching algorithm in this paper converts 8-bit pixel values to 1-bit ones of 0 or 1 based on the mean pixel value in the search block, and performs motion estimation. The image simplification through bit conversion reduces computational loading for motion estimation and enables fast searching. Moreover, adaptive searching methods according to the motion information in the block make the procedure of motion estimation fast and exact by eliminating unnecessary searching processes of low motion block and deepening a searching procedure in high motion block.

The rest of this paper is organized as follows. In Section 2 and 3, the proposed algorithms are described. In Section 4, its simulation performance is evaluated, and conclusions are given in Section 5.

II. Block Matching Algorithm

Block matching is the process of obtaining motion vectors. In the FS-BMA, the present frame of an image sequence is divided into subsequent non-overlapping $N \times N$ blocks. For each block in the frame, the most similar block to it in the search area of the reference frame is chosen as the matched block. The Figure 1 shows the block matching process between two frames. The blocks to be searched in the reference frame are within a search area of size $(N+2w) \times (N+2w)$, where w is the maximum displacement that a block can move between two frames. The number of search locations for one motion vector is $(2w+1)^2$. At each search location, the displaced block difference (DBD) is computed by

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



(a) Reference Frame (b) Current Frame

Fig. 1. Full search block matching method

where F_n is the block being predicted by motion

estimation in the present frame, F_{n-1} is a candidate block within the search area in the previous frame, and N indicates the width and height of the block. Among the candidate blocks in the search area, the block that has the minimum value of DBD is selected as the best matched block, and the corresponding displacement (x, y) in the horizontal and vertical directions becomes the motion vector.

1. Block matching algorithm based on bit-patterns

Bit patterns mean binary images that represent the 8-bit pixel values in the block and search area as 1-bit ones [5]. This aims to achieve a fast searching speed by reducing the amount of computation in the searching process. Given a block of size $N \times N$, the bit-pattern Bs of the search block and search area are simply calculated by

$$B(i,j) = \begin{cases} 1, & if\ p(i,j) \geq M \\ 0, & otherwise \end{cases} \quad (2)$$

where $p(i, j)$ represents the (i, j)th pixel value in the search block or search area, and M represents the mean pixel value of the search block. The bit-patterns represent intuitive block information on the search block and search area. That is, if the bit-pattern of the search block is very similar to that of a certain block in the search area, the two blocks in the original images is supposed to have similar image characteristics. There, however, a lot of prediction errors occur, so the methods to reduce them should be followed. In case of full searching based on the bit-pattern, 83% of the computation amount can be reduced in the DBD calculation, comparing with that in the FS-BMA. The reason is that image expression is simplified to one bit from 8 bits. The equation to obtain the numbers of operation for one DBD calculation is as

$$N(n) = 32n + \sum_{k=0}^{7} 2^k (n+7-k)/8 \quad (3)$$
$$= 32n + (255n + 247)/8$$

For example, in case of 1 bit(n=1) and 8 bit(n=8), the operation numbers are $N(1)$=95 and $N(8)$=542, respectively [6]. In this paper, the proposed block matching process based on bit pattern is as follows.

* Step 1 : Generates the bit-patterns on the search block and search area
* Step 2 : Compares the bit-pattern of the search block with that of each candidate block in the search area
* Step 3 : Saves top of m numbers of motion vectors of candidate blocks that have the most numbers of corresponding bits
* Step 4 : Computes DBDs for the saved candidates blocks, and selects the block that has the minimum of DBD among them as the matched block

What the value of m is affects the computation amount and prediction errors. Figure 2 shows the example of block matching based on bit-patterns.

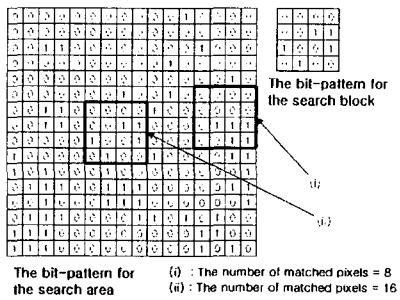


Fig. 2. Block matching based on bit-patterns

III. Adaptive block matching according to the motion information in blocks

Performing consistent motion estimations on every block is not an effective way in terms of the speed and the

exactness in prediction. In order to predict exact motion in a short time, adaptive prediction methods according to the motion information in blocks are required.

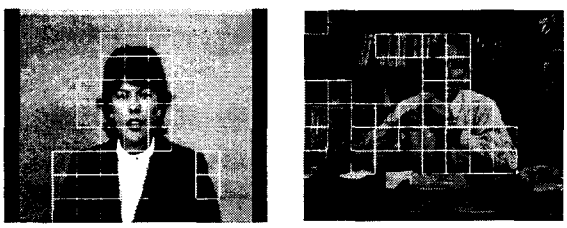
1. Decision of no-motion blocks

Generally, each frame of motion pictures consists of the object part that has motion and the background part that has no motion. If the motion estimation process on the background part is performed the same as that on the motion object part, many unnecessary searching processes may occur. Therefore, whether it has motion or not has to be checked before the motion searching begins. This can be decided by setting the motion vector at $(0,0)$ as in Equation 4 and calculating DBD.

$$DBD(0,0) = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |F_n(i,j) - F_{n-1}(i,j)|$$

$$\begin{cases} if \ \ DBD(0,0) \le TH_{no-motion}, & No\ motion\ searching \\ otherwise, & Motion\ searching \end{cases} \quad (4)$$

If DBD(0,0) is smaller than a certain threshold, the block is supposed to has no motion, and it can be excluded in the process of motion prediction. Like this, its heavy computational loading can be reduced with eliminating unnecessary matching processes. For the blocks proved to have motion, the motion estimation is performed according to the degree of motion. Figure 3 shows the formation of the blocks with motion on the Claire and Salesman images. You can see that the number of blocks is less than 1/3 of the number of blocks to be formed on the whole image. This means that motion estimation is not performed on more than 2/3 of the images.



(a) Claire (b) Salesman

Fig. 3. Decision of blocks with motion

2. Motion estimation using variable-block size

Variable block size method is used to perform more exact motion estimation in the blocks that has many prediction errors. When the DBD with the matched block is bigger than a certain threshold, the search block is partitioned by

$$\begin{cases} if \ \ DBD(i,j) \ge TH_{split}, & Split \\ otherwise, & Not\ split \end{cases} \quad (5)$$

Block matching for each divided block is performed again, and then, when the DBD for each divided block is bigger than the threshold, the block is redivided. Figure 4 shows block partitioning levels

(B_{S0}, B_{S1}, B_{S2}, and B_{S4}) and their block sizes.

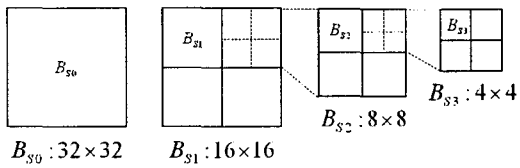
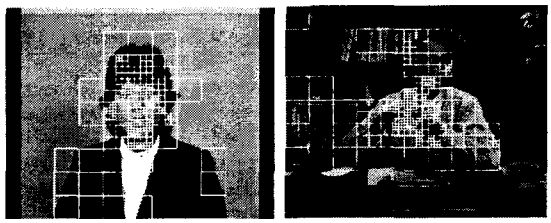


Fig. 4. Block formation according to block levels

Such partitioning method means that the matching process deepens in the part that has many prediction errors (that is, in the part that has big motion in the image). It reduces the prediction errors much so that it can improve the image quality, while it corresponds effectively to the regional motion change of image. Figure 5 shows the variable-size block formation according to the motion information in the Claire and Salesman images.



(a) Claire (b) Salesman

Fig. 5. Block partitioning according to the motion information

3. Setting the secondary search area

In case the motion vector of the predicted block is $(\pm w, \pm w)$ of the maximum displacement, it indicates the boundary of the search area. This means the real motion vector can indicate outside of the search area. If the matched block is within, not boundary, the search area, the matching process can be terminated or continued with partitioning the block according to the DBD. However, when the motion vector indicates the boundary of the search area, a secondary search area, which size is 1/2 of the original search area, is set and the block searching is performed again [7]. That is, the first search area can be set in small size to reduce the computational loading, and in case of big motion the secondary search area is set for the matched block to be found in it.

In this paper, the secondary search area method is used in the lower block levels such as B_{S1}, B_{S2}, and B_{S3}. Figure 6 shows the process. The motion vector (MV) is obtained by adding the two motion vectors as in Equation 6.

$$MV = MV_{first} + MV_{second} \tag{6}$$

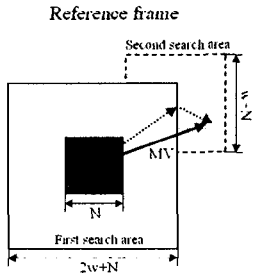


Fig. 6. The second search area

4. Setting the location of search area, subsampling of pixels, and size of search area

Search area should be set in the place that has a high possibility of matched block being. In most image sequences, motions are smooth and slowly varying. Therefore, the discontinuity of motion vectors occurs only at the boundary of objects moving in different directions. Generally, as motion objects exist over several blocks, motion vectors between adjacent blocks are highly correlated. If the location of search area using such a correlation is set, it is helpful in making the motion estimation efficient. Figure 7 shows the previously obtained motion vectors of the adjacent blocks.

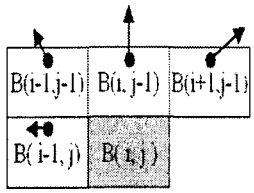


Fig. 7. Motion vectors of the neighbor blocks

With the motion vectors of adjacent blocks, we calculate 4 DBDs with the search block and a zero-motion DBD at its location. Then, we set the location of the search area with the motion vector that has the minimum DBD. It is only used in the case of the top-level block B_{S0}. In the case of lower block levels B_{S1}, B_{S2} and B_{S3}, the motion vector that is obtained from its higher block level is used. Therefore, in spite of setting a small search area, big motion vectors can be obtained by the inheritance of motion vectors from higher block levels.

In FS-BMA, all pixels are participated in measuring the degree of similarity among candidate blocks in search area. But, if pixel subsampling is adapted, considering a high spatial correlation between adjacent pixels according to the degree of motion in blocks, the computational complexity can be reduced. On the other hand, if excessively high rates of subsampling are appliec, it reduces the accuracy of prediction, especially when the motion of an image sequence is high. Therefore, it is important that the subsampling rates of pixels are decided properly according to the degree of motion in blocks. That block size gets smaller means that the degree of motion gets bigger. So, as it goes to the lower block level, the lower subsampling rate of pixels should be set, which prevents the simplification of bit pattern in the low block

level. Figure 8 shows the examples of subsampling patterns that can be applied.

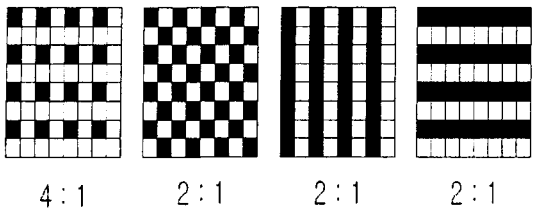


Fig. 8. Examples of pixel subsampling

The subsampling rates of pixels according to block levels in this study are as follows.

* B_{S0} : 4:1 pixel subsampling
* B_{S1} : 2:1 pixel subsampling
* B_{S2}, B_{S3} : No pixel subsampling

The size of search area is also set according to the degree of motion in blocks, as the subsampling of pixels. As it goes to the lower block level that has bigger motion, the bigger size of search area is set, as shown in Equation 7.

$$w_{S0} \le w_{S1} \le w_{S2} \le w_{S3} \qquad (7)$$

Which W_{S0}, W_{S1}, W_{S2}, and W_{S3} are the maximum displacements that blocks can move according to each block level.

5. Final motion vector

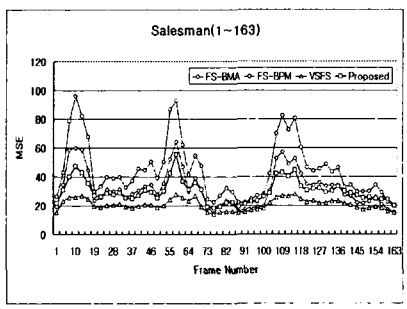
The motion vectors in each block level are obtained from Equation 6. So, the final motion vector is obtained by adding them, as shown in Equation 8.

$$MV_{final} = MV_{S0} + MV_{S1} + MV_{S2} + MV_{S3} \qquad (8)$$

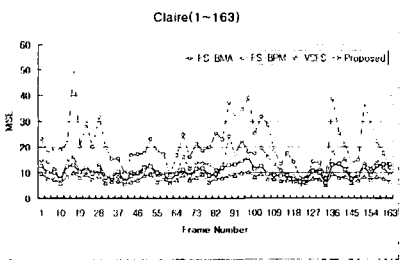
IV. Results of experiments and consideration

In simulations, we use from No. 1 to No. 163 frames of the Claire image (CIF: 352×288) and No. 1 to No.163 frames of the Salesman image (CIF: 352×288) in order to evaluate the performance of algorithms. Motion estimation is performed every three frames by applying the reference frame period of general video coding. The basic block size of the proposed algorithm is 32×32 and it can be divided up to 4×4. The size of search area is set to increase the maximum displacement $W_{S0} = 1$, $W_{S1} = 2$, $W_{S2} = 3$, and $W_{S3} = 4$ as the level of block became lower. Considering the secondary search area, the maximum motion vector is ± 14. The comparative algorithms are the FS-BMA, FS-BPM(Full Search Bit Pattern Matching) and VSFS(Variable Size Full Searching). The basic block size of the algorithms is 16×16, and the maximum motion vector is set to be $\pm 14 (w = 14)$. And the VSFS can be divided up to 4×4. For the comparative analysis of algorithms, the Mean Square Error(MSE) is obtained through simulations, and the Peak Signal to Noise Ratio(PSNR) is calculated as a distortion measure.

Table 1 and Figure 9 show the comparison between the proposed algorithm and the comparative algorithms. As you can see from Table 1, the proposed algorithm has improved approximately 0.5dB of PSNR with fewer numbers of blocks per frame than other algorithms. Figure 9 shows that the proposed algorithm occurs fewer numbers of errors per frame and reduces the error deviation between frames.



(a) Claire



(b) Salesman

Fig. 9. MSE comparison for the Claire and Salesman CIF sequences

operations. The reason is that the representation of image sequence has been transformed 8-bit pixel values into 1-bit, the blocks with no motion have been excluded from the matching processes, and then, the size of search area and the pixel subsampling rates have been set according to the degree of motion. In the propcsed algorithm, the inheritance of motion vectors from top to lower levels has made the value of big motion vector in spite of setting small search area, which, most of all, has taken a great role in reducing the computational loading.

Table 1. The performance evaluation

	Algorithm	MSE/pixel	PSNR	Relative number of operations	Number of blocks
Claire (CIF)	FS-BMA	12.19068	37.27052	1	396
	FS-BPM	21.31385	34.84325	0.337985	396
	VSFS	7.696624	39.2678	1.08418	562.4182
	Proposed	10.44131	37.94325	0.010608	178.6364
Salesman (CIF)	FS-BMA	32.35502	33.03139	1	396
	FS-BPM	44.77563	31.62039	0.337985	396
	VSFS	20.86094	34.93747	1.123066	628.3091
	Proposed	29.93309	33.36929	0.016875	297.1818

The computation complexity of algorithms is evaluated by calculating the number of operations for each algorithm [8]. In order to process one pixel, three operations (subtraction, absolute-value calculation, and addition) are required. In case of the FS-BMA, when the block size is $16\times16(=256)$ and the size of the search area is $(2w+1)^2=841(w=14)$ $841\times(256\times3+1)$ operations are required to get a motion vector on one block; number 1 is the comparative operation with the minimum DBD after the DBD calculation. In short, the number of operations required to estimate one frame of CIF image (396 blocks) is

$$841\times(256\times3+1)\times396=256,104,684.$$

When the number of operations of the FS-BMA is 1, the relative numbers of operations of the algorithms are presented in Table 1. As shown in Table 1, the proposed algorithm represents relatively the very least number of

V. Conclusions

In this paper, we have proposed a new block matching algorithm that causes fewer prediction errors and performs much faster motion estimation than the FS-BMA by using the variable block size and many kinds of adaptive matching methods based on bit-patterns. The proposed algorithm reduces the computation loading of the FS-BMA resulted from the consistent motion searching not considering motion information in blocks. Which can be done because the proposed algorithm performs a adaptive matching according to the degree of motion based on bit-patterns and eliminates, if possible, unnecessary matching processes. As for many regional motions of image that cannot be treated effectively when the block size is fixed, the block partitioning is tried. Simulation results have proved that the proposed algorithm provides better performance, average 0.5dB PSNR improvement and about 99% savings in the number of

operations in case of $w = 14$, than the FS-BMA.The proposed algorithm is worth applying to motion estimation for low bit rate video compression such as video-telephone or video-conference, which have low motions within images *and require fast real-time processing.*

The proposed algorithm has applied adaptive methods in most parts of the matching processes, but the threshold value is used the same in all frames, which is not effective because the degree of motion can vary according to frames. *If different thresholds are obtained according to the motion* information of each frame, better results can be expected. Therefore, before performing motion estimation, the studies of the preprocessing for obtaining the optimal threshold should be followed.

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] S. Kappagantula, and K. R. Rao, "Motion compensated interframe image prediction," IEEE Trans. Communication, Vol. COM-33, pp. 1011-1015, 1985.

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

[4] J. Zhang, M. O. Ahmad, and M. N. S. swamy, "A New Variable Size Block Motion Compensation," Proceedings of the IEEE Int. Conf. on Image Processing, Vol. 2, pp. 164-167, 1997.

[5] J. Feng, K.-T. Mehrpour, and A.E. F. Karbowiak, "Adaptive block matching algorithm for video compression," IEEE Proc. Vision, Image and Signal Processing, Vol. 145, No. 3, pp. 173-178, 1998.

[6] H. Kiya, J. Furukawa, and Y. Noguchi, "Block Matching Motion Estimation Based on Median Cut Quantization for MPEG Video," IEICE Trans. *Electronics Communications & Computer Sciences*, Vol. E82-A, No. 6, pp. 899-904, 1999.

[7] 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.

[8] J. Chalidabhongse, and J. Kuo, "Fast motion vector estimation using multiresolution spatio- temporal correlations," IEEE Trans. Circuits and Systems for Video Technology, Vol. 7, No. 3, pp. 477-488, 1997.

권 혁 봉(Heak-Bong Kwon) 정회원

1992년 : 호서대학교 정보통신공학과 졸업(공학석사)

2001년 : 충북대학교 정보통신공학과 졸업(공학박사)

1997년 ~ 현재 : 김포대학 전자정보계열 조교수

<관심분야> : 영상신호처리, 컴퓨터 비젼, 디지털 신호처리

송 영 준(Young-Jun Song) 정회원

1996년 : 충북대학교 정보통신공학과 졸업 (공학석사)

1996년 ~ 1998년 : LG전자 멀티미디어 사업본부 주임연구원

1998년 ~ 2000년 : LG반도체 메모리사업본부 M/M 개발팀 주임 연구원

2000년 ~ 2003년 : 한국전자통신연구원 네트워크연구소 네트워크팀 선임연구원

2003년 ~ 현재 : 충북대학교 정보통신공학과(박사과정)

<관심분야> : 영상인식, 영상처리, 얼굴 인식