블럭 정합 알고리즘을 위한 적응적 비트 축소 MAD 정합 기준과 VLSI 구현

(An Adaptive Bit-reduced Mean Absolute Difference Criterion for Block-matching Algorithm and Its VlSI Implementation)

오황석[†] 백윤주[††] 이흥규[†††]
(Hwang-Seok Oh) (Yunju Baek) (Heung-Kyu Lee)

요 약 블럭 정합 알고리즘의 VLSI 구현시 복잡도를 줄이고, 수행 속도를 높이기 위하여 새로운 정합 기준인 적응적 비트 축소 MAD(adaptive bit-reduced mean absolute difference:ABRMAD)를 제안한다. ABRMAD는 기존의 MAD에서 화소의 모든 비트를 비교하는 대신, 화소를 구성하는 중요한 비트만을 고려하여 축소된 화소 값을 비교하여 움직임 벡터를 찾는다. 실험을 통하여, 제안한 정합 기준은 기존의 MAD 정합 기준에 비하여 낮은 하드웨어 복잡도를 가지면서 MSE(mean square error) 측면에서 유사한 성능을 가짐을 보인다. 또한 기존의 비트 수 축소형 정합 기준인 DPC(difference pixel counting), BBME(binary-matching with edge-map), 그리고 BPM(bit-plane matching)과 비교하여 같은 수의 비트를 사용하였을 경우 좋은 MSE 성능을 가짐을 보인다.

Abstract An adaptive bit-reduced mean absolute difference(ABRMAD) is presented as a criterion for the block-matching algorithm(BMA) to reduce the complexity of the VLSI implementation and to improve the processing time. The ABRMAD uses the lower pixel resolution of the significant bits instead of full resolution pixel values to estimate the motion vector(MV) by examining the pixels in a block. Simulation results show that the 4-bit ABRMAD has competitive mean square error(MSE) results and a half less hardware complexity than the MAD criterion. It has also better characteristics in terms of both MSE performance and hardware complexity than the Minimax criterion and has better MSE performance than the difference pixel counting(DPC), binary block-matching with edge-map(BBME), and bit-plane matching(BPM) with the same number of bits.

1. Introduction

The motion estimation(ME) techniques have an important role in video coding due to their capabilities that reduce the temporal redundancies residing between successive frames. Most of the ME techniques developed so far are the block-matching algorithms(BMA) which estimate a motion vector (MV) on a block-by-block basis. In the BMA, the current frame is partitioned into non-overlapping equal-size rectangular blocks and all pixels in one block are assumed to have the same motion displacement. The MV for each block is estimated by searching for its highest correlation with an associated block within the search area in the reference frame. A general search algorithm is the full search BMA(FSBMA), which obtains the optimal result by searching exhaustively for the best matching block within a search window. Many of video coding standards such as H.261[1], H.263[2], and MPEG-1,2[3] have adopted the ME technique.

· 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

[†] 비 회 원 : 한국전자통신연구원 가상현실연구개발센터 연구원
hsoh@etri.re.kr
[††] 비 회 원 : 숙명여자대학교 정보과학부 교수
yunju@sookmyung.ac.kr
[†††] 종신회원 : 한국과학기술원 전산학과 교수
hklee@casaturn.kaist.ac.kr

논문접수 : 1998년 8월 17일
심사완료 : 2000년 2월 23일

The enormous amount of computational demand for the ME, especially FSBMA, prevents the software implementation from running timely in real-time video coding systems. Therefore, the VLSI implementation of the ME is indispensable for such systems. Owing to the regularity and simplicity of matching operations, the FSBMA is suitable for the VLSI implementation[4,5,6] and thus it is the most widely used for the ME.

There are three factors that determine the performance of the BMA: 1) search method, 2) search range, and 3) block-matching criteria. Many search methods have been investigated in the literatures[7, 8,9,10]. Among them, the full search method gives an optimal solution and has low control overheads. The block-matching criterion adopted in a search method affects the accuracy of the estimation. Since the matching is a main operation of the VLSI architecture for the BMA, a hardware-efficient matching criterion is highly essential. Thus several block-matching criteria to reduce hardware requirements have been investigated[11,12,13,14,15]. In this paper, we focus on the block-matching criterion to propose a new scheme for reduction of the hardware requirement and speed-up of the VLSI implementation with an acceptable video performance.

As the estimation in the BMA has been done with block-by-block comparison which is performed in pixel-by-pixel, the basic operation of the BMA is a pixel comparison. Since the pixel comparison is a bitwise operation, if the number of bits is reduced, the corresponding hardware requirement can be reduced. Taking this point into consideration, we propose a more hardware-efficient block-matching criterion, called adaptive bit-reduced mean absolute difference(ABRMAD).

The rest of the paper is organized as follows. In Sec. 2, various well-known block-matching ME criteria and some lower bit resolution criteria are briefly discussed before presenting the ABRMAD criterion. In Sec. 3, we present a new criterion ABRMAD for the BMA. In Sec. 4, the performance of the ABRMAD is evaluated and compared with other methods such as MAD, MiniMax, DPC, BBME, and BPM. Finally, the hardware characteristics synthesized by using VLSI design tools are presented in Sec. 5.

2. Criteria for Block-Matching Motion Estimation

Many BMA criteria have been presented to find the best matching block among the candidate blocks in the search area. The normalized cross-correlation function(NNCF)[7], mean-square difference(MSD)[8], and mean absolute difference(MAD)[9] are well known criteria. Because the NNCF and MSD require the multiplication, both of them are too complex and their hardware realization seems far from being feasible. Thus the MAD criterion is widely used for the ME and it has been implemented with VLSI. But the hardware implementation of the MAD with full resolution of pixels is still computationally expensive. Therefore, new matching criteria, such as the difference pixel counting(DPC)[12], binary block-matching with edge-map(BBME)[13], and bit-plane matching(BPM)[14], have been presented with lower resolution of the pixel to reduce the hardware complexity of the MAD. In this section, we briefly describe the MAD and several criteria that use pixel data of the lower resolution.

Assume $R(i,j)$ to be pixel intensity at (i,j) position of the current block in the current frame, and $S(i+u,j+v)$ to be pixel intensity at (i,j) location of the candidate block in the reference frame, shifted by the u pixels and v lines within the search area. For the best match, the displacement (u,v), called MV, represents the estimates of the displacement in horizontal and vertical directions, respectively. In the following criteria, the size of block is NxN and the position (u,v) lies within the search area.

(1) Mean absolute difference(MAD)[9]:

$$\mathrm{MAD}(u,v)=\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}|R(i,j)-S(i+u,j+v)|. \quad (1)$$

In this criterion, the MV is determined by the smallest MAD(u,v) for all possible displacements (u,v) within the search area. The MAD is the most widely used matching criterion due to its lower computational complexity.

(2) Minimized maximum error(MiniMax)[11]:

$$\mathrm{MiniMax}(u,v) = |R(i,j) - S(i+u, j+v)|_{\max}, \tag{2}$$

where $1 \le i, j \le N$. In this recently developed method, the smallest MiniMax(u,v) within the search area is chosen for the best match. Chen et al.[11] reported that the MiniMax criterion requires 15% less hardware than the conventional MAD criterion while it maintains an acceptable video performance.

(3) Difference pixel count criterion(DPC)[12]:

The difference pixel count criterion(DPC) is defined as the number of pixels whose 2-bit requantized values are matched. It can be described as follow.

$$\mathrm{DPC}(u,v) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{AND}\{\mathrm{XNOR}(R_k(i,j), S_k(i+u, j+v))\}, \tag{3}$$

where $R_k(i,j)$ and $S_k(i+u,j+v)$ are the requantized values of $R(i,j)$ and $S(i+u,j+v)$, respectively. The requantized pixel resolution is lower than original one, thus the DPC can require less hardware than the MAD. The requantized value $P_k(i,j)$ is obtained as eq. (4).

$$P_k(i,j) = \begin{cases} 11 & \text{if } p(i,j) - \overline{m} \ge t \\ 10 & \text{if } t > p(i,j) - \overline{m} \ge 0 \\ 01 & \text{if } 0 > p(i,j) - \overline{m} \ge -t \\ 00 & \text{if } p(i,j) - \overline{m} < -t, \end{cases} \tag{4}$$

where $p(i,j)$ is pixel intensity at position (i,j), and $\overline{m}$ denotes the mean of the block. For each block, the value t used in eq. (4) can be derived using

$$t = \frac{3}{2N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} |p(i,j) - \overline{m}|. \tag{5}$$

(4) Bit-plane matching criterion(BPM)[13]:

In this method, the current and the reference frames are transformed into binary-valued pixel frames. For two binary blocks, the matching criterion, called BPM, is as follow.

$$\mathrm{BPM}(u,v) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=0}^{N} \mathrm{XOR}(\hat{R}(i,j), \hat{S}(i+u, j+v)), \tag{6}$$

where $\hat{R}(i,j)$ and $\hat{S}(i+u,j+v)$ denote pixel values after the transformation to one-bit frames. The MV is defined by the (u,v) for which BPM(u,v) is minimum. To transform the current and reference frames, they apply the convolution kernel K to the original frame F to obtain the filtered version G. The pixels of the binary frame $\hat{F}$ are given by

$$\hat{F}(i,j) = \begin{cases} 1 & \text{if } F(i,j) \le G(i,j) \\ 0 & \textit{otherwise}. \end{cases} \tag{7}$$

They use the convolution kernel K, for example, given by

$$K(i,j) = \begin{cases} \frac{1}{25} & \text{if } i,j \in [1,4,8,12,16] \\ 0 & \textit{otherwise}. \end{cases} \tag{8}$$

(5) Binary block-matching using the edge-map(BBME)[14]:

The BBME estimates the MV by using the binary edge-map to reduce the complexity of hardware implementation. For the comparison of the current block R and a candidate block S, the binary edge-maps of the block $\overline{R}$ and $\overline{S}$ are compared. The edge-maps of the current frame and the reference frame are obtained by using the pre-processing. This method can reduce the complexity of hardware because of using one-bit per pixel for pixel comparison. The matching criterion is as follow.

$$\mathrm{BBME}(u,v) = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{XOR}(\overline{R}(i,j), \overline{S}(i+u, j+v)). \tag{9}$$

The MV is the displacement which is pointed to by the smallest BBME(u,v).

(6) Reduced bit mean absolute difference (RBMAD)[15]:

Any n-bit pixel value A, let $A_{p:q}$ be $p-q+1$ bits $A_p \cdots A_q$ in the binary representation of $A = A_{n-1} \times 2^{n-1} + \cdots + A_1 \times 2^1 + A_0$ where $n \ge p \ge q \ge 0$. The RBMAD criterion is described as follow.

$$\mathrm{RBMAD}_k(u,v) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} |R(i,j)_{n-1:n-k} - S(i+u, j+v)_{n-1:n-k}|, \tag{10}$$

where k is a given factor which should be $n \ge k > 0$. Similar to MAD, the MV is determined by the smallest RBMAD$_k(u,v)$ for all possible displacements (u,v) within the search area. Since the RBMAD uses only upper k bits of pixels, the RBMAD is equivalent to the MAD when $k=n$. This criterion can reduce hardware complexity depending on k but causes the significant degradation of video quality.

3. An Adaptive Bit-Reduced Mean Absolute Difference Criterion

The criterion RBMAD is very effective to reduce the hardware complexity and to improve the processing speed. But the degradation of video quality is a critical problem. The video quality of the RBMAD applied for video sequences that contain large dark area, that is, pixels have effective values at close to the least significant bit(LSB), is degraded seriously. To overcome the problem of the performance degradation, if only effective bits are compared by examining the pixels of the block, the significant degradation of video quality can be remedied. Taking this point into consideration, an efficient matching criterion, ABRMAD, is proposed.

In the ABRMAD, for any n-bit pixel A, selected k bits are represented as $A_{m:m-k+1}$ similar to the RBMAD. But the most significant bit position m is obtained by examining the current block in contrast to the RBMAD where m is set to the most significant bit(MSB) of the pixel(i.e., $m=n-1$).

The criterion of ABRMAD is described as follow.

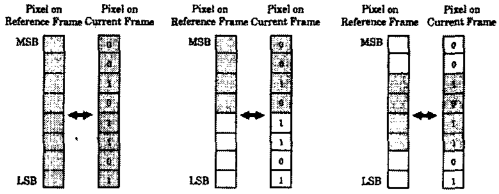
$$\mathrm{ABRMAD}_{m,k}(u,v)=$$

$$\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|R(i,j)_{m:m-k+1}-S(i+u,j+v)_{m:m-k+1}\right|, \qquad (11)$$

where $k(n\geq k>0)$ represents that how many bits are used and m is the effective MSB position in the range of $n-1\geq m>0$. Similar to the RBMAD, the MV is determined by the smallest $\mathrm{ABRMAD}_{m,k}(u,v)$ for all possible displacements (u,v).

The effective MSB position m is obtained by looking at the current block. For pixels in the block, the non-zero MSB position of the pixel that has the largest intensity is set to m. In case of $k-1\leq m$, the k bits from m to $m-k+1$ bit position are used. In other case($k-1>m$), the k bits from $k-1$ to 0 bit position are used. The number of bit(bit width) k is pre-determined by considering the required quality and hardware complexity.

An example for pixel comparisons of the criteria MAD, RBMAD with 4 bits, and ABRMAD with 4 bits is shown in Figure 1, where the largest pixel value of the current block is $45(00101101_2)$, $n=8$, and $k=4$. Thus the effective most significant bit position m is set to 5. Therefore, the 4-bit pixel values positioned from 5 down to 2, instead of 8-bit pixel values, are compared in the ABRMAD. In MAD criterion, all n bits are used to compare the pixels and 4 MSB bits are used in the RBMAD. If $k=n$, the ABRMAD is the same to the MAD and the RBMAD.



(a) (b) (c)

Fig. 1 (a) Example of selected bits for pixel comparison using MAD (b) Pixel comparison using the RBMAD with 4 bits (c) Pixel comparison using ABRMAD with 4 bits

4. Experimental Results

Four quarter common intermediate format(QCIF: 176×144, 30 frames/s) video sequences, named *Miss America*, *Carphone*, *Foreman*, and *Claire* are used to investigate the performance of the ABRMAD criterion. Two video sequences, *Miss America* and *Claire*, contain a speaker with slow movements, which are typical in low bit-rate video applications such as video-phone, video-conferencing. The sequences *Carphone* and *Foreman* have a little moderate motion field which is caused by the movements of the objects and the effects of the video camera. The experimental results are compared with those of existing ME criteria such as MAD, MiniMax, RBMAD, BPM, DPC, and BBME.

The mean square error(MSE) per pixel is used for a performance measure of the ME criteria. The MV for a block of size 16×16 is estimated within the ±16 allowable displacement.

Table 1 Effective MSB position of each test sequence in ABRMAD

Effective MSB position	Miss America	Carphone	Fpreman	Claire
Bit position 0	0%	0%	0%	0%
Bit position 1	0%	0%	0%	0%
Bit position 2	0.8%	0%	0%	0.2%
Bit position 3	3.9%	0.6%	0.3%	5.9%
Bit position 4	11.7%	1.1%	14.1%	12.7%
Bit position 5	26.3%	29.0%	21.8%	24.0%
Bit position 6	44.0%	31.9%	30.1%	38.4%
Bit position 7	13.2%	37.4%	33.7%	18.8%

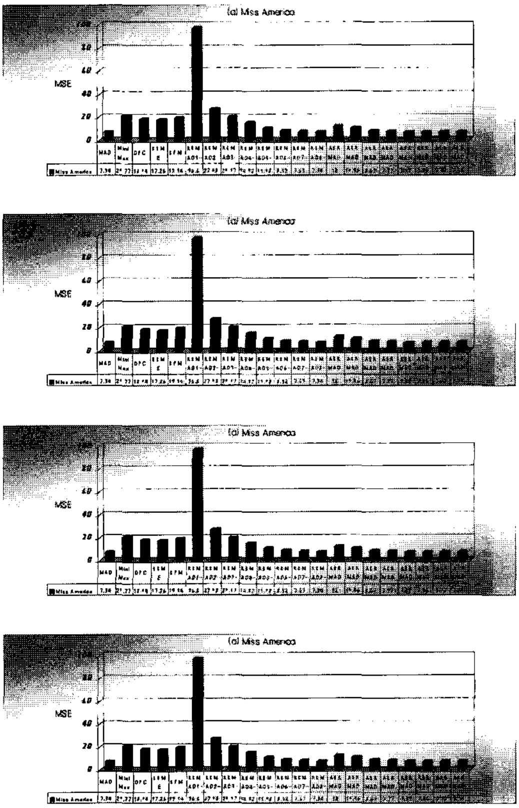


Fig. 2 Average MSE performance of each matching criterion

The average performance of each matching criterion is summarized in Figure 2. The ME for the matching criteria is performed within only luminance components. The numbers of bits, used for ME, of DPC, BBME, and BPM are 2, 1, and 1 bit(s), respectively. The motion compensated frame is used as a reference frame. The MSE performances of RBMAD and ABRMAD are described according as how many bits are used to compare the pixels in a block. The two matching criteria RBMAD and ABRMAD which use 8 bits are the same to the MAD. The MSE performances of the ABRMAD with small number of bits such as 5, 4 bits are similar or slightly worse compared with those of the MAD. But it is better than those of the RBMAD with the same number of bit and MiniMax. In particular, from the results of the video sequences

Miss America and *Claire*, the MSE performances of the ABRMAD with small bits are very similar to those of MAD. The results also show that the MSE performances of the ABRMAD with one or two bits are better than those of the DPC, BBME, and BPM. In the video sequence with low motion fields, the performance gains are not great, but in the sequence *Carphone* and *Foreman*, the MSE performance of the ABRMAD with one bit is much better than those of BBME and BPM. Also, the ABRMAD with two bits has good MSE performance comparing with DPC. In Table 1, the percentages of effective MSB position of each video sequence are shown. The sequences *Miss America* and *Claire* have the bit position 6 as the most occurring effective MSB position. It show that the ABRMAD can have a change to improve the MSE performance comparing with RBMAD.

The pure MSE performance of the proposed ABRMAD shows that it is a good ME criterion for low complexity hardware.

5. VLSI design

The VLSI synthesis for the conventional MAD, RBMAD, and the proposed ABRMADhave been implemented by Compass VLSI design automation tools using the architecture described in [4]. They are described and tested with a high-level hardware description language, Verilog and its simulator. Then, the layout for the synthesis is automatically generated by the ASIC compilers. Similarly, the timing aspect of the synthesis is tested using a critical path analyzer in the tools.

The VLSI design for the ABRMAD consists of a effective MSB finder to detect the effective MSB, two bit-selectors, and processing elements(PEs) for estimating the MV as shown in Figure 3. The effective MSB finder finds out the largest pixel value and sets the effective MSB position, which can be implemented by using multiplexer, counter, and shift-register. Two bit-selectors pass k bits of pixels from the effective MSB position m to $m-k+1$ in the current block and the search area. The bit-selector can be implemented by using shift-register. The PE architecture compares the current block with the candidate block and generates the MV. The VLSI structure described in [4] is used for the PE architecture(as shown in Figure 4) which is for a 8×8 current block. Each PE element calculates the sum of absolute differences of the corresponding pixels between the current block and a candidate block. In the figure, the index CB denotes sequences of current block data, SWL and SWR denote sequences of reference frame data from the different portions of the search area. The pixels in the left a half portion and right a half portion of the search area are passed through the SWL and SWR, respectively. The pixels in the current block are passed through from PE0 to PE7 at each cycle time. The pixels in the search area are broadcasted to all PEs. The detailed data-flow diagram and its features are described in [4]. Each PE element consists of an accumulator for accumulation of sum of the absolute differences, an absolute difference component which is composed with a subtracter and an absolute function to calculate the difference of the two pixels. The PE architecture can be easily scaled to perform the ME with a large size of blocks.

The structure for the ABRMAD is similar to the RBMAD except a single comparator as shown in Figure 3. The overhead for the ABRMAD compared with the RBMAD is so minor that it can be ignored. But as shown in Figure 2, the MSE performance gain is very high. With the same video quality to MiniMax, the ABRMAD can be implemented with a quite less VLSI complexity.

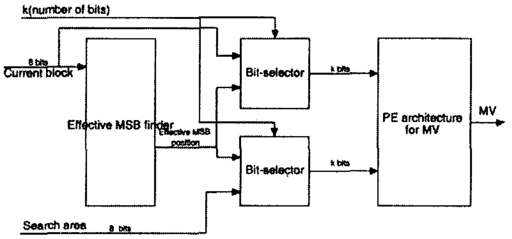


Fig. 3 Block diagram for VLSI architecture of ABRMAD

With a 0.8 ㎛ CMOS standard cell technology, the characteristics of the synthesized VLSI design are shown in Table 2. The table shows that the less bits we use, the smaller hardware cost and the faster speed we can achieve. Comparing with RBMAD with the same bit, the ABRMAD requires more hardware caused by the effective MSB finder and bit-selectors by 1-2% of the MAD criterion. The speed degradation, also, is 1-3% of the MAD criterion. This drawback compared with RBMAD can be compensated by MSE performance of the ABRMAD. Examining the Figure 2 and Table 2, the trade-off between video quality and hardware cost in the ABRMAD can be observed. As an example, take a look at the case of the 4-bit ABRMAD. With negligible degradation of video performance, we can implement the ME with 56% smaller hardware area and 33% faster speed than that using MAD criterion. Thus, if cost-effective real-time applications are required, the presented criterion is very suitable for the VLSI implementation of the ME.

6. Conclusion

A new ME criterion called ABRMAD which is very suitable for an efficient VLSI implementation without a significant degradation of video quality is presented. Two main advantages of the proposed

Table 2 Performance table for VLSI implementation(Area:mil^2, Speed:cycle time(ns))}

Criterion	RBMAD				ABMAD			
	Area	AreaBRMAD AreaMAD	Speed	SpeedBRMAD SpeedMAD	Area	AreaBRMAD AreaMAD	Speed	SpeedBRMAD SpeedMAD
MAD	148.82×97.35	100%	32.06	100%	148.82×97.35	100%	32.06	100%
7-bit	138.49×95.30	91.1%	29.47	91.9%	139.70×96.23	92.8%	30.01	93.6%
6-bit	126.52×88.28	77.1%	27.04	84.3%	127.73×89.12	78.6%	27.52	85.8%
5-bit	112.66×74.13	57.6%	24.97	77.9%	113.94×74.87	58.9%	25.05	78.1%
4-bit	100.06×62.17	42.9%	21.14	65.9%	102.03×62.21	43.8%	21.49	67.0%
3-bit	87.72×46.60	28.2%	19.09	59.5%	89.03×47.11	29.0%	19.61	61.2%
2-bit	78.27×47.35	25.6%	17.43	54.4%	79.58×48.16	26.5%	17.91	55.9%
1-bit	55.72×31.02	11.9%	17.22	53.7%	56.91×31.92	12.5%	17.33	54.1%

ABRMAD criterion are 1) VLSI area is much saved, 2) operational speed is increased. In order to show the hardware efficiency of the proposed scheme, we synthesized the ME using the ABRMAD criterion and compared it with the statistics of the conventional MAD criterion. Through intensive simulation tests, we also showed that the ABRMAD has competent video performance like MAD. As a result, the new scheme can be adopted in VLSI implementation of real-time video applications.

References

[1] M. Liou, "Overview of the Px64 kbit/s Video Coding Standard," *Comm. ACM*, vol. 34, no. 4, pp. 59-63, 1991.

[2] R. Schaphorst, "Draft Recommendation H.263(Video coding for low bitrate communication)," *ITU-T SG15 WP15/1*, 1995.

[3] D. L. Gall, "MPEG: A video compression standard for multimedia application," *Comm. ACM*, vol. 34, no. 4, pp. 46-58, 1991.

[4] K. Yang, M. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," *IEEE Trans. Circ. Syst.*, vol. 36, no. 10, pp. 1317-1325, 1989.

[5] T. Komarek and P. Pirsch, "Array architecture for block-matching algorithms," *IEEE Trans. Circ. Syst.*, vol. 36, no. 10, pp. 1301-1308, 1989.

[6] L. D. Vos and M. Stegherr, "Parameterizable VLSI architectures for the full-search block-matching algorithm," *IEEE Trans. Circ. Syst.*, vol. 36, no. 10, pp. 1309-1316, 1989.

[7] H. G. Musmann, P. Pirsch, and H. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, no. 4, pp. 523-548, 1985.

[8] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Comm.*, vol. 29, no. 12, pp. 1799-1806, 1981.

[9] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *Proc. NTC81*, pp. G5.3.1-G5.3.5, 1981.

[10] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Comm.*, vol. 33, no. 8, pp. 888-896, 1985.

[11] M. -J. Chen, L. -G. Chen, T. -D. Chiueh, and Y. -P. Lee, "A new block-matching criterion for motion estimation and its implementation," *IEEE Trans. Circ. Syst. Video Tech.*, vol. 5, no. 3, pp. 231-236, 1995.

[12] S. Lee, J. M. Kim and S. -I. Chae, "New motion estimation using low-resolution quantization for MPEG2 video encoding," *Proc IEEE Workshop on Signal Processing*, pp. 428-437, 1996.

[13] M. M. Mizuki, U. Y. Desai, I. Masaki, and A. Chandrakasan, "A binary block-matching architecture with reduced power consumption and silicon area requirement," *Proc. ICASSIP*, pp. 3248-3251, 1996.

[14] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *IEEE Trans. Circ. Syst. Video Tech.*, vol. 7, no. 4, pp. 702-706, 1997.

[15] Y. Baek, H. -S. Oh, and H. -K. Lee, "New block-matching criterion for efficient VLSI implementation of motion estimation," *Electro. Lett.*, vol. 32, no. 13, pp. 1184-1185, 1996.

[16] *Computer arithmetic: principles, architectures, and design*, John Wiley & Sons. 1979.

[17] *Telenor H.263 software,* Telenor reaearch at http://www.nta.no/Brukere/DVC/H263_software/.

[18] H. Gharavi and M. Mills, "Block matching motion

estimation algorithm - new results," *IEEE Trans. Circ. Syst.*, vol. 37, no. 5, pp. 649-651, 1990.

오황석

1992년 2월 경북대학교 자연과학대학 전자계산학과 졸업(학사). 1994년 2월 한국과학기술원 전산학과 졸업(공학석사). 1999년 3월 한국과학기술원 전산학과 졸업(공학박사). 1999년 1월 ~ 현재 한국전자통신연구원 가상현실연구개발센터(선임연구원). 관심분야는 video coding, parallel processing for video coding, multimedia system

백윤주

1990년 한국과학기술대학 전산학과 졸업(공학 학사). 1992년 한국과학기술원 전산학과 졸업(공학 석사). 1997년 한국과학기술원 전산학과 졸업(공학 박사). 1997년 ~ 1998년 (주) 핸디소프트 선임연구원. 1998년 ~ 1999년 한경대학교 컴퓨터공학과 전임강사. 1999년 ~ 현재 숙명여자대학교 정보과학부 조교수. 관심분야는 멀티미디어 시스템, 전자상거래

이홍규

1978년 서울대학교 전자공학과 졸업. 1981, 1984년 한국과학기술원 전산학 석사, 박사. 1984년 ~ 1986년 미시간대학교 연구연구원. 1986년 ~ 현재까지 한국과학기술원 전산학과 교수. 관심분야는 실시간 고장허용 처리, 영상처리, 원격탐사등