A NEW BLOCK-MATCHING ALGORITHM BASED ON AN ADAPTIVE SEARCH AREA ADJUSTMENT USING SPATIO-TEMPORAL CORRELATION

Hwang-Seok Oh[1], Choong-Hoon Lee[2], Heung-Kyu Lee[2], and Joon-Hyeon Jeon[3]
[1]Visual Information Research Team, Virtual Reality Research and Development Center,
Electronics and Telecommunications Research Institute(ETRI),
161 Kajeong-dong Yusong-gu, 305-350, Taejon, Korea
email: hsoh@etri.re.kr
[2]Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST),
373-1, Kusong-Dong, Yusong-Gu, Taejon, 305-701, Korea
email: {chlee,hklee}@casaturn.kaist.ac.kr
[3]Internet Business Division, Communications Service Planning Center, Korea Telecom,
206 Chongja-dong, Pundang-gu, Songnam city, Kyonggi-do 463-711, Korea
email: memory@kt.co.kr

Abstract— **The motion estimation & compensation technique is widely used for video coding applications but the real-time motion estimation is not easy due to its enormous computations. In this paper, a new adaptive reduction of search area for the block-matching algorithm is presented to reduce the computational complexity of the full search block-matching algorithm for low bit-rate video coding. The proposed method exploits the correlation of successive video frames and adjusts the size of search area depending on the displaced block difference and the block classification information of the block in the previous frame. Simulation results show that the proposed algorithm has similar mean square error performance to the full search block-matching algorithm but only requires less a half computational complexity than the full search algorithm.**

Keywords : Block-matching algorithm, Motion estimation, Video coding

I. INTRODUCTION

The Motion estimation & compensation technique has been widely used in video compression due to its capability of reducing the temporal redundancies between frames[1], [2]. Most of the algorithms developed for motion estimation so far are block-based techniques, called block-matching algorithm(BMA). In this technique, the current frame is divided into fixed size of blocks, then the each block is compared with candidate blocks in reference frame within the search area. The widely used approach for the BMA is the full search BMA(FSBMA), which examines all candidate blocks within the search area to obtain a motion vector(MV). The MV is a displacement between the block in current frame and the best-matched block in reference frame in horizontal and vertical directions.

Because of the intensive computation of the FSBMA, it is hard to implement the video codec with software only for real-time processing. Many researches have been studied for real-time processing of video coding applications. One of them is the VLSI implementation for the motion estimation[3], [4], [5]. It is frequently used in complicated or full-motion video coding applications for real-time processing. The other approach is a development of fast search algorithms. These techniques, (1) fast BMA with unimodal error surface assumption[6-17], (2) fast BMA with pixel subsampling[13], [14], [18], (3) fast BMA with reduced search area[19], [20], [21], [22], [23], and (4) hierarchical BMA[24], [25], have been investigated.

The most frequently used fast BMA with unimodal error surface - the matching error increases monotonically as the search point moves away from position of the global minimum - is reported in many literatures[6-17]. These methods, for example, three step search(TSS)[6], new three step search(NTSS)[7], 2-D logarithm search(2-DLOG)[8], parallel hierarchical 1-D search(PHODS)[12], and four step search(FSS)[16], etc., have a critical problem, which is falling into a local-minimum. This problem significantly degrades the video quality as the encoding process is being progressed. Another widely used approach for fast BMA, reducing the search area depending on the motion fields, are studied in [19], [20], [21], [22], [23]. Lee et al.[19] proposed the method of reducing the size of search area in TSS according to the magnitude of displaced block difference(DBD). The adaptive BMA(ABMA)[20] adjusts the search window depending on the DBDs between a current block and candidate blocks pointed to by the MV of its neighborhood blocks. The large search window is used for the block having a large DBD and the small search area is allocated to the block having a small DBD, respectively. This method was simple, but ignored the facts that low bit rate video sequences have center-biased motion distri-

Manuscript received June 28, 1999

bution and the correlation between DBD and MV magnitude is not significant. The method, called dynamic adjustment of the search window with fixed/variable size of blocks(DASWF/V)[22], [23] are presented to adjust the search window using block similarity with fixed and variable size of blocks. These methods have some problems to be implemented on real-time video codec. The DASWF/V used the spatial correlation but have huge overhead for preprocessing to get block similarity and block segmentation. Fan et al.[21], proposed the motion estimation algorithm based on the structure segmentation. However, the methods in[21], [22], [23] are too complex to be implemented for real-time processing and the schemes in [19], [20] do not predict motion structures well, since only the DBD is considered as an indicator of motion fields.

In this paper, a new BMA with adaptive adjustment of search area is presented to reduce the computational complexity by exploiting the spatio-temporal correlations in video sequences. Especially the proposed scheme intends for low bit-rate video coding applications such as video-phone and video-conferencing. The video frames of these applications contain only one or more speaker and invariant backgrounds. The number of search points can be reduced by setting a small size of search area for a block within the invariant background regions and large one for a block in moving regions. The paper organized as follows. In Section II, some characteristics of low bit-rate video sequences got by computer experiments and the BMA with a new search area adjustment scheme are presented. Simulation results with several test video sequences are shown to evaluate the proposed scheme in Section III.

II. MOTION ESTIMATION ALGORITHM

The proposed motion estimation algorithm is performed with variable size of search area depending on block types. Based on the analysis of the low bit-rate video sequences, the current frame is classified into four types corresponding to the stationary background, active moving block, and changing block from background to active region and vice versa. First, some characteristics of video sequences of low bit-rate applications are examined, then an adaptive search area adjustment scheme is presented.

A. Characteristics of low bit-rate video sequences

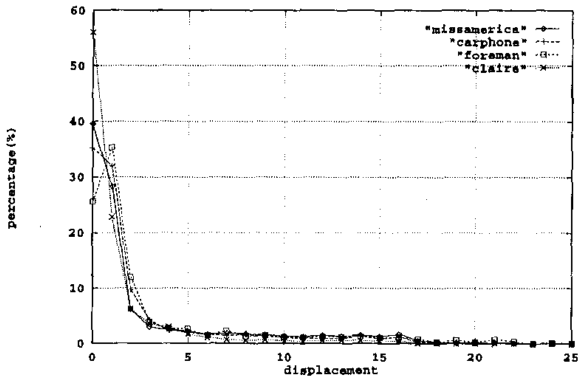


Fig. 1. The MV distribution of each video sequence. The values of X-axis is the the euclidean distances of MVs

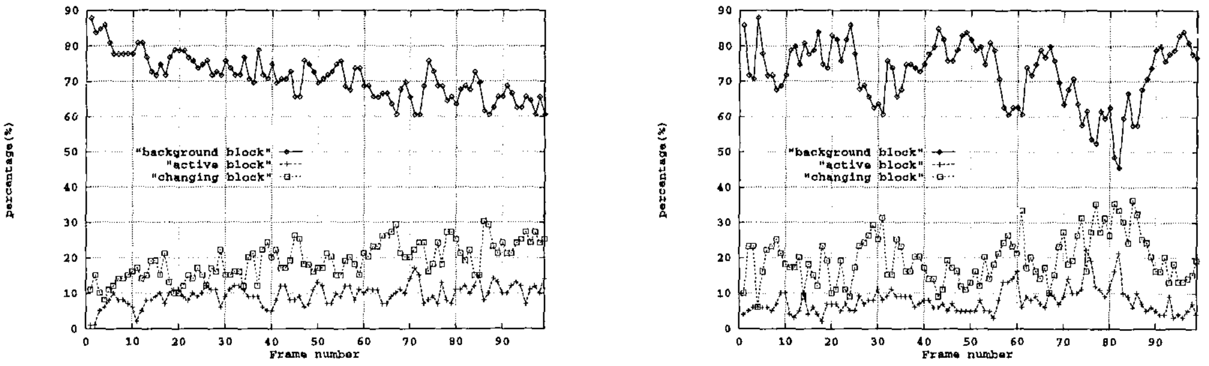
The video sequences for low bit-rate video coding applications such as video-phone, video-conferencing have some restrictive motion characteristics. Through various simulations in the previous works[7], [19], [20], [21], [22], [23], it is found that these video sequences are gentle, smooth, and vary slowly. Thus, the MV distribution of the best-matched block is center-biased. The MV distributions of the simulation results which are performed by using the FSBMA with search area ±16 of each video sequence are plotted in Fig. 1. It shows that the MVs lie in near search origin(displacement < 5) by 90% or so. From the figure, we can see that only a few blocks have large motion displacement, and lots of computation can be reduced if these blocks are selected from the other blocks

In addition to that, these video sequences have a significant correlations between successive frames because they usually contain only a speaker with slow motion field which is caused by movement of the speaker. Thus, a block in specific region in previous frame can belong to the same region at that position in current frame: a block in background region may lie in background region in current frame. The percentages of each type of blocks in successive frames are shown in Fig. 2. In all video sequences, the percentage of background block in successive frame is very high. The active blocks and changing blocks occupy only 30% below. Also, the pattern of distribution is very similar without regard to video sequences. It shows that the temporal correlation between successive frames is very high, that is, if a block in previous frame belongs to background region or active region, the block which is located in the same position in current frame may be classified into background block or active moving block, respectively, with a strong probability.

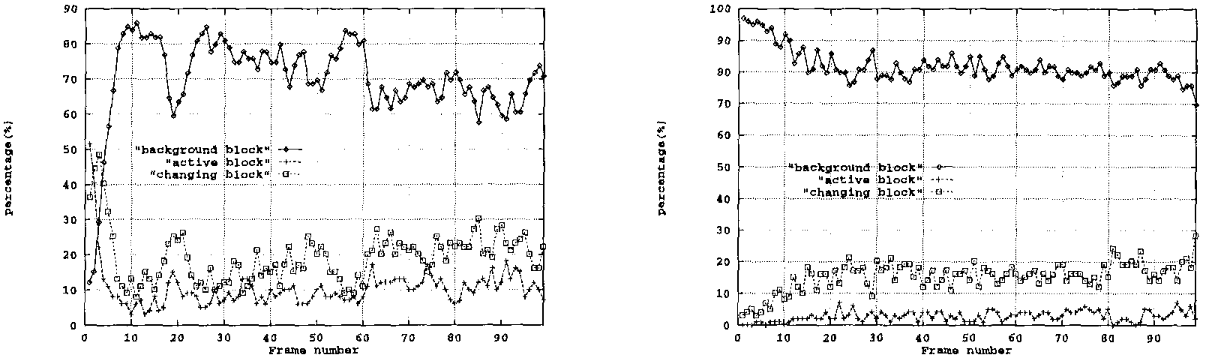
Based on the these facts, many fast search algorithms, for instance, NTSS, FSS, etc., have been developed. These methods improve the motion estimation performance both in terms of computational complexity and video quality.

B. Motion estimation with adjustment of search area

A new BMA with dynamic reduction of search area to cut down the number of search points by using the char-



(a) Miss America (b) Carphone



(c) Foreman (d) Claire

Fig. 2. *The percentage ratio of the background, active and changing blocks*

acteristics of video sequences of low bit-rate applications is presented : small search area is assigned to the background block and large one for the active moving block.

In the proposed scheme, to determine the size of search area, a block in the current frame is classified whether it lies in the background or in the active regions which are moving regions of objects. The block is classified into one of the four types : background block, active moving block, changing block from background to active region and vice versa. For the background block and the changing block from active region to background, small search area can be assigned, and for the active moving block and the changing block from background to active region, large search area is required. To classify the block, we use the block classification information(BCI) of the block in previous frame and DBD at current frame.

The proposed motion estimation is performed through the following steps: (1) Estimate the initial values of thresholds to determine the BCI of a given block; (2) Set the size of search area by using the DBD and the BCI, and then perform the BMA within the determined search area; and (3) Update the thresholds for an adaptation to the video sequences' characteristics.

B.1 Estimate the initial thresholds

Three parameters, $BCI(i)$, DBD_{bg}, and DBD_{ar} for block classification are used. $BCI(i)$ is the block classification information of i^{th} block in a frame, DBD_{bg} is a mean displaced block difference of blocks which are classified into the background block in previous frame, and DBD_{ar} is a mean displaced block difference of blocks which are classified into the active block in previous frame. The initial values of these parameters are estimated by using results of first frame by means of the FSBMA. In the first frame, the $BCI(i)$ is determined by only the MV. The blocks that have zero displaced MV are classified into background and the other blocks are classified into active blocks. The DBD_{bg} is set to the mean of DBD of background blocks and the DBD_{ar} is set to the mean of DBD of active blocks. The DBD_{bg} is usually small compared with DBD_{ar}. These parameters are changing to reflect adaptively the characteristic of video sequences as the coding process is progressed.

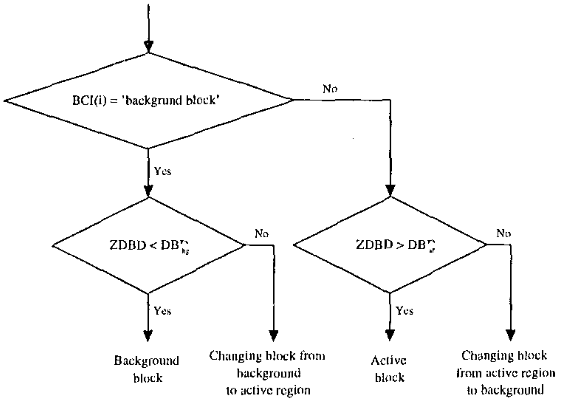


Fig. 3. Block diagram for block classification with block classification information of the previous frame and DBDs

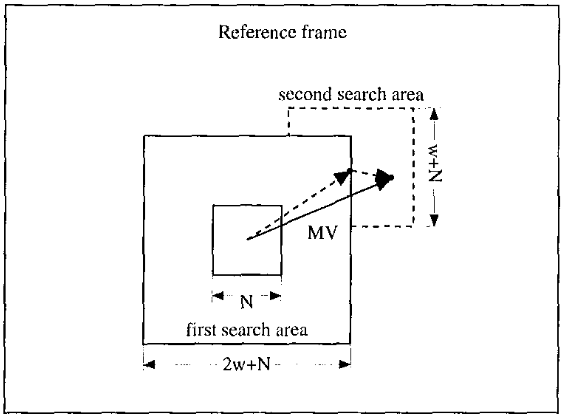


Fig. 4. An example of motion estimation with the determined search area

B.2 Set the size of search area and perform the BMA

The classification of blocks in current frame is done through the use of $BCI(i)$ of the previous frame, DBD_{bg}, and DBD_{ar}. The size of search area is determined depending on the BCI of the block. For a given block in the current frame, the block is classified as shown in Fig. 3. In the figure, the $BCI(i)$ is the block classification information for i^{th} block in previous frame, and then DBD_{bg} and DBD_{ar} are the mean DBDs of background blocks and active blocks in previous frame, respectively. The ZDBD is a zero displaced block difference of the given block in current frame. To determine the BCI of i^{th} block in current frame, firstly, the algorithm observes the BCI at that position in previous frame. If the $BCI(i)$ at the previous frame is 'background block', then, the ZDBD in current frame is compared with DBD_{bg} to decide whether the block is classified into the background block or the changing block from background to active region. If the $BCI(i)$ is 'active block', then, the block can be classified into the active block or the changing block from active region to background depending on ZDBD and DBD_{ar}.

For each block type, the search area is set as w for the active block and the changing block from background to active region, $w/4$ for background block, and $w/2$ for the changing block from active region to background, respectively, where w is an initial maximum displacement of the MV. That is, the larger search area is allocated for active region that contains high-motion blocks and the smaller search area is given for background blocks.

Once the size of search area is determined, the FS-BMA is performed. If the best-matched candidate block positioned within search area, the motion estimation is stopped. Otherwise(MV points to the boundary of search area), the size of search area becomes a half of first search area and then the FSBMA is performed. Above procedure continues until the MV does not point to the boundaries of search area.

Fig. 4 shows the procedures to find the MV. In first step, the MV points to the boundary of first search area, therefore, motion estimation goes on with second search area. The motion estimation process stops in the second search area because the second MV points to the position within the search area. Final MV can be obtained by using vector summation of the first and second MV. When the motion estimation process continues and the size of search area becomes 1, the next search area is set to 1 and motion estimation is performed continuously until the MV points new search origin. This process can reduce the quality degradation caused by misclassification of block types.

B.3 Update the thresholds

After a frame goes through the BMA, the BCI of each block is updated by using of BCI of the block at the same position in current frame, which is used for next frame. At the same time, the DBD_{ar} and DBD_{bg} are reestimated by using the DBDs of the background blocks and the active blocks of current frame.

III. SIMULATION RESULTS

Four QCIF(quarter common immediate format:176 × 144) and CIF(common immediate format:352 × 288) video sequences, called *Miss America, Carphone, Foreman,* and *Claire,* are used to evaluate the proposed search area adjustment scheme with the block size of 16 by 16. The initial allowable displacements of search area are set to ±8 and ±16. The mean absolute error(MAE) criterion is used for a matching criterion. Each scheme for adjustment of search area is applied to the full search(FS). The test video sequence *Miss America* and *Claire* have a speaker with slow movement and the sequence *Foreman* and *Carphone* have moderate motion displacements with some change of backgrounds. The motion estimation is performed only in the luminance component of the video sequences. The results of proposed method are compared with those of conventional one-FSBMA- which set the size of search area to initial value and those of ABMA.

The average performance of the algorithms is summarized in TABLE I in terms of the mean square error(MSE) per pixel and the number of matching points(NSP) per block. The NSP is the number of candidate blocks which are compared with the current block to obtain the best-matching block. In general, the NSP per block is $(2w+1)^2$, where w is initial displacement. But the blocks at the frame boundary have a smaller NSP than others. Therefore, the NSP is calculated by counting the number of blocks which are actually compared.

When each scheme is applied to the FS, the proposed scheme has better MSE performance than ABMA with similar or less complexity. It has also similar MSE performance compared with the FSBMA but only requires about a half or a third of NSP. The MSE performance gain is not quite good in simulation with test sequences that have slow motion fields such as *Miss America* and *Claire*. But the NSP reduction is noticeable. Comparing with the FSBMA and ABMA, only 25% and 50% of NSP are required, respectively. With video sequences which have moderate motion fields such as *Carphone* and *Foreman*, the large MSE gain is obtained with having less than a half NSP in comparison with ABMA. In some cases, ABMA shows a good MSE performance but it has same or similar NSP to the FSBMA since the motion classification scheme failed. By simulation results, the proposed method for search area adjustment is compatible with the FSBMA but requires only a half or less computation complexities, especially in the applications of low bit-rate video transmission. In ABMA, the results are not good because the update method of thresholds for motion classification does not work well. As shown in TABLE I, the MSE performance and NSP are varying according to the test video sequences. Especially in low motion video sequences such as *Miss America* and *Claire*, the reduction of NSP is not significant and the MSE performance does not good in comparison with the proposed method.

The figures from 5 to 8 present the MSE performance with the *Miss America* and *Foreman* sequences. The figures show that the MSE performance of the presented scheme is similar to that of FSBMA along the frame number and is more efficient than that of ABMA. The method for search area adjustment can be used in combination with fast search algorithm such as TSS, NTSS, PHODS, etc. to get more reduction of complexity. It makes real-time processing of video encoding possible in the areas of interactive video applications.

IV. CONCLUSION

A new adjustment scheme of the search area is proposed based on the DBD and the block classification information for low bit-rate video coding. The algorithm is very simple and efficient in terms of MSE performance and computational complexity. The method can be easily adapted to fast search algorithms which are developed so far such as TSS, 2-DLOG, PHODS, etc. to reduce computational complexity quite more.

REFERENCES

[1] D. L. Gall(1991) MPEG: A video compression standard for multimedia application. *Communication of the ACM*, **34**(4):46–58.
[2] M. Liou(1991) Overview of the Px64 kbit/s video coding standard. *Communication of the ACM*, **34**(4):59–63.
[3] K. Yang, M. Sun & L. Wu(1989) A family of VLSI designs for the motion compensation block-matching algorithm. *IEEE Transactions on Circuits and Systems*, **36**:(10):1317–1325.
[4] T. Komarek & P. Pirsch(1989) Array architecture for block matching algorithms. *IEEE Transactions on Circuits and Systems*, **36**(10):1301–1308.
[5] L. D. Vos & M. Stegherr(1989) Parameterizable VLSI architectures for the full-search block-matching algorithm. *IEEE Transactions on Circuits and Systems*, **36**(10):1309–1316.
[6] J. R. Jain, & A. K. Jain(1981) Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, **Com-29**(12):1799–1808.
[7] R. Liu, B. Zeng, & M. L. Liou(1994) A new three-step search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **4**(4):438–441.
[8] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, & T. Ishiguro(1981) Motion-compensated interframe coding for video conferencing. *Proc. NTC81*, pp. G5.3.1–G5.3.5.
[9] R. Srinivasan & K. R. Rao(1985) Predictive coding based on efficient motion estimation. *IEEE Transactions on communications*, **33**(8):888–896.
[10] M. -J. Chen, L. -G. Chen & T. -D. Chiueh(1994) One-dimensional full search motion estimation algorithm for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, **4**(5):504–509.
[11] M. Ghanbari(1990) The cross-search algorithm for motion estimation. *IEEE Transactions on Communications*, **38**(7):950–953.
[12] L. -G Chen, W. -T Chen, Y. -S Jehng, & T. -D. Chiueh(1991) An efficient parallel motion estimation algorithm for digital image processing. *IEEE Transactions on Circuits and Systems for Video Technology*, **1**(4):378–385.
[13] B. Liu & A. Zaccarin(1993) New fast algorithms for the estimation of block motion vectors. *IEEE Transactions on Circuits and Systems for Video Technology*, **4**(2):148–157.
[14] B. -M. Wang, J. -C. Yen & S. Chang(1994) Zero waiting-cycle hierarchical block matching algorithm and its array architectures. *IEEE Transactions on Circuits and Systems for Video Technology*, **4**(1):18–28.
[15] E. Chan, A. A. Rodringuez, R. Gandhi & S. Panchanathan(1994) Experiments on block-matching techniques for video coding. *ACM Multimedia Systems*, **2**(5):228–241.
[16] L. -M. Po & W. -C. Ma(1996) A novel four-step search algorithm for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **6**(3):313–317.
[17] H. Jozawa, K. Kamikura, A. Sagta, H. Kotera & H. Watanabe(1997) Tow-stage motion compensation using adaptive global MC and local affine MC. *IEEE Transactions on Circuits and Systems for Video Technology*, **7**(1):75–85.
[18] Y. -L. Chan & W. -C. Siu(1996) New adaptive pixel decimation for block motion vector estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **6**(1):113–118.
[19] L. -W Lee, J. -F Wang, J. -Y. Lee, & J. -D. Shie(1993) Dynamic search-window adjustment and interlaced search for block-matching algo-

TABLE I

AVERAGE MSE AND NSP OF EACH SEARCH AREA ADJUSTMENT SCHEME APPLIED FOR THE FS

Video format	Displacement	BMA scheme	Test sequences							
			Miss America		Carphone		Foreman		Claire	
			MSE	NSP	MSE	NSP	MSE	NSP	MSE	NSP
QCIF	8	FSBMA	7.45	236	70.50	236	102.43	236	9.27	236
		ABMA	8.80	180	72.83	132	105.80	135	9.49	136
		Proposed	7.46	68	71.24	104	103.59	106	9.30	60
	16	FSBMA	7.34	886	67.71	886	94.92	886	9.17	886
		ABMA	8.30	574	68.74	455	100.03	467	9.44	477
		Proposed	7.40	225	68.22	334	95.79	340	9.25	192
CIF	8	FSBMA	7.37	262	42.57	262	39.10	262	5.13	262
		ABMA	7.82	216	43.37	194	43.49	144	5.15	242
		Proposed	7.41	78	43.43	150	39.22	156	5.21	84
	16	FSBMA	7.26	984	28.32	984	29.64	984	5.10	886
		ABMA	7.75	432	29.06	443	31.05	410	5.12	462
		Proposed	7.32	302	28.55	376	29.87	385	5.11	265

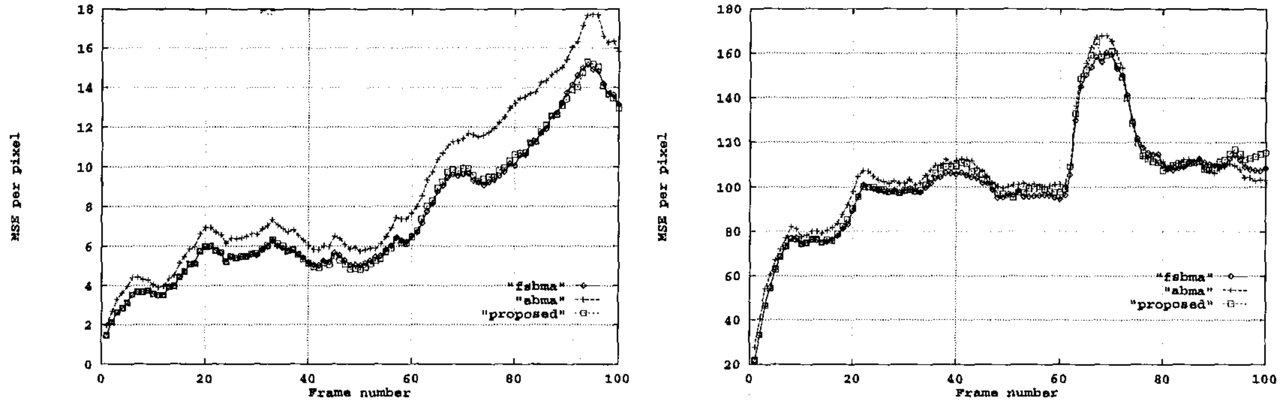


Fig. 5. MSE performance for the *Miss America* and *Foreman* QCIF sequences with block size of 16 × 16 and displacement ±8

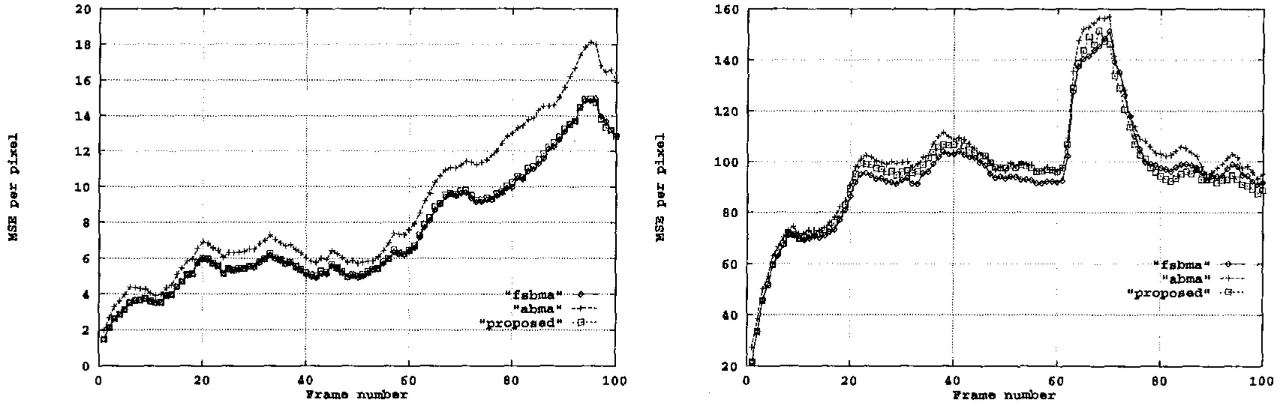


Fig. 6. MSE performance for the *Miss America* and *Foreman* QCIF sequences with block size of 16 × 16 and displacement ±16

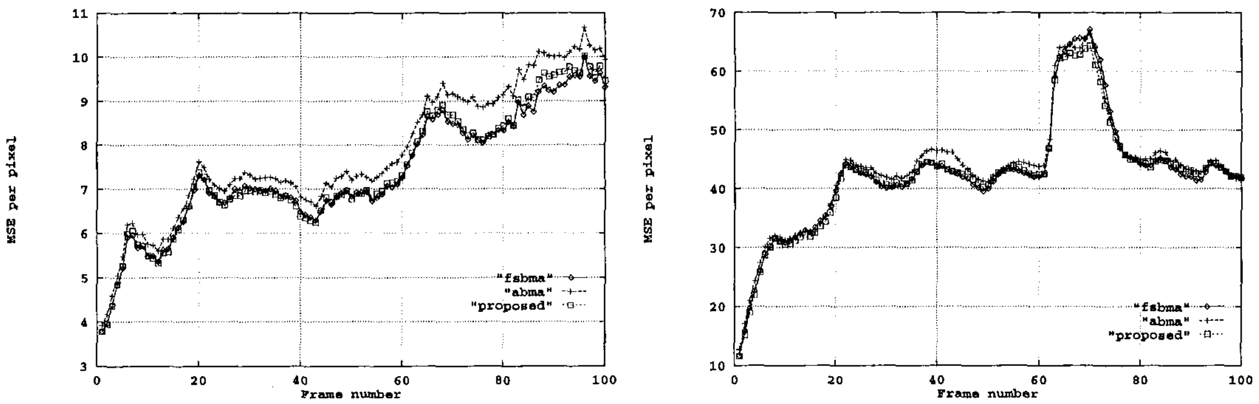


Fig. 7. MSE performance for the *Miss America* and *Foreman* CIF sequences with block size of 16 × 16 and displacement ±8

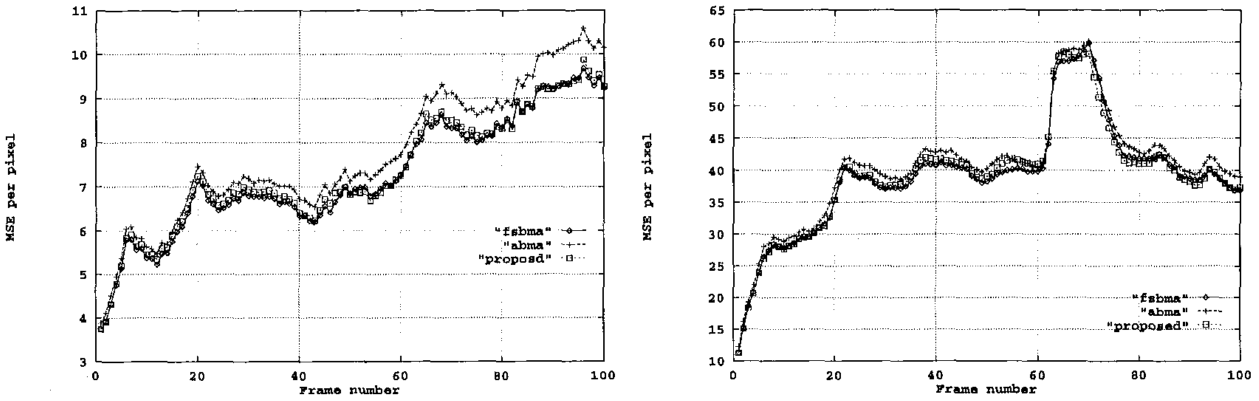


Fig. 8. MSE performance for the *Miss America* and *Foreman* CIF sequences with block size of 16 × 16 and displacement ±16

rithm. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(1):85–87.

[20] J. Feng, K.-T. Lo, H. Mehrpour & A.E. Karbowiak(1995) Adaptive block matching motion estimation algorithm for video coding. *Electronics letters*, **32**(18):1542–1543.

[21] J. Fan, J. Yu, X. Xue, & L. Wu(1998) Efficient motion estimation algorithm based on structure segmentation and compensability analysis. *Optical Engineering*, **37**(5):1563–1570.

[22] H. -S. Oh & H. -K. Lee(1998) Block-matching algorithm with dynamic search window adjustment for low bit-rate video coding. *Journal of electronic imaging*, **7**(3):571–577.

[23] H. -S. Oh & H. -K. Lee(1998) Adaptive adjustment of the search window for block-matching algorithm with variable block size. *IEEE transactions on consumer electronics*, **44**(3):659–666.

[24] M. Bierling(1988) Displacement estimation by hierarchical block matching. *Proc. SPIE Visual Communications and Image Processing'88*, 1001:942-951.

[25] G. Gupta & C. Chakrabarti(1995) Architectures for hierarchical and other block matching algorithm. *IEEE Transactions on Circuits and Systems for Video Technology*, **5**(6):477-489.

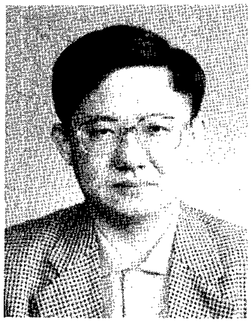
[26] V. Seferidsi & M. Ghanbari(1993) General approach to block-matching motion estimation. *Optical Engineering*, **32**(7):1464–1474.

[27] W. Li & E. Salari(1995) Successive elimination algorithm for motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **4**(1):105–107.

[28] M. -J. Chen, L. -G. Chen, T. -D. Chiueh & Y. -P. Lee(1995) A new block-matching criterion for motion estimation and its implementation. *IEEE Transactions on Circuits and Systems for Video Technology*, **5**(3):231–236.

Hwang-Seok Oh received the BS degree in computer science from Kyungpook National University, Korea, in 1992 and his MS and PhD degree in Computer Science from Korea Advanced Institute of Science and Technology(KAIST), Korea, in 1994, 1999, respectively. He is a member of research engineer at Virtual Reality Research and Development Center, Electronics and Telecommunications Research Institute(ETRI) in Korea since 1999. He is a member of IEEE, ACM, and SPIE. His research interests include video coding, multimedia systems, VLSI architectures for signal processing and content based information retrieval.

Choong-Hoon Lee received the B.S. degree in Computer Engineering from the Dongguk University, Seoul, Korea in 1996 and his MS degree in Computer Science from Korea Advanced Institute of Science and Technology(KAIST), Korea, in 1998. He is currently Ph.D candidate in Computer Science, KAIST. His research interests are video and image coding, video parsing, digital watermarking and multimedia systems.



Heung-Kyu Lee received the B.S. degree in electronics engineering from the Seoul National University, Seoul, Korea in 1978, and the MS and PhD degree in Computer Science from the Korea Advanced Institute of Science and Technology(KAIST), in 1981, 1984, respectively. From 1984 to 1985 he served as a Post-Doc. in the University of Michigan, Ann Arbor. In 1986 he joined the Korea Advanced Institute of Science and Technology as a professor. Since 1990, he has been a research staff at the Satellite Research Center in KAIST. He is a member of IEEE and ACM. His major interests are VLSI and parallel processing, fault-tolerant and real-time computing, multimedia systems.

Joon-Hyeon Jeon received the B.S. degree from DongKoog University, Korea , in 1984, and the M.S. and Ph.D. degrees from Korea Advanced Institute of Technology , Korea, in 1984 and 1991, respectively. He is Team Leader at Internet Service Development Team of Korea Telecom Multimedia Research Lab., Seoul Korea, where his current research interests are in the area of video service development on the Internet. He was at Image & Information Laboratory in Korea Telecom from 1991 to 1998, where he was engaged in research in image compression and coding such as MPEG1, MPEG2, MPEG4 and H26x