논문 2010-47SD-12-7

H.264 비디오 코덱을 위한 효율적인 움직임 추정 알고리즘과 회로 구조

(Efficient Motion Estimation Algorithm and Circuit Architecture for H.264 Video CODEC)

이 선 영*, 조 경 순**

(Seonyoung Lee and Kyeongsoon Cho)

요 약

본 논문은 H.264 비디오 코덱에 적용할 수 있는 고성능 정수화소 움직임 예측 회로 구조에 대해 설명한다. 전역 탐색 알고리즘은 모든 가능한 블록에 대해 확인하기 때문에 가장 좋은 결과를 보장한다. 그러나 전역 탐색 알고리즘은 많은 양의 연산과 데이터를 요구한다. 연산 노력을 줄이기 위해 많은 고속 탐색 알고리즘들이 제안되었다. 고속 탐색 알고리즘들의 단점은 데이터 접근이 불규칙하고 데이터 재사용이 어려운 것이다. 본 논문에서는 고성능 움직임 예측을 위하여 효율적인 정수화소 움직임 예측 알고리즘을 제안하고 있으며, 이를 구현하기 위한 처리 속도가 높고 외부 메모리 사용을 줄일 수 있는 회로 구조를 제안한다. 제안한 회로는 7가지 종류의 가변 블록 크기를 지원하면 41개 움직임 벡터를 생성한다. 구현된 고성능 움직임 예측 회로는 RTL로 구현하였고 FPGA가 탑재된 보드에서 동작을 검증하였다. 130nm CMOS 표준 셀 라이브러리로 합성된 회로는 1초에 139.8장의 1080HD (1,920x1,088) 영상을 처리할 수 있고 H.264 5.1 레벨까지 지원 가능하다.

Abstract

This paper presents a high-performance architecture of integer-pel motion estimation circuit for H.264 video CODEC. Full search algorithm guarantees the best results by examining all candidate blocks. However, the full search algorithm requires a huge amount of computation and data. Many fast search algorithms have been proposed to reduce the computational efforts. The disadvantage of these algorithms is that data access from or to memory is very irregular and data reuse is difficult. In this paper, we propose an efficient integer-pixel motion estimation algorithm and the circuit architecture to improve the processing speed and reduce the external memory bandwidth. The proposed circuit supports seven kinds of variable block sizes and generates 41 motion vectors. We described the proposed high-performance motion estimation circuit at RTL and verified its operation on FPGA board. The circuit synthesized by using 130nm CMOS standard cell library processes 139.8 1080HD (1,920x1,088) image frames per second and supports up to H.264 level 5.1.

Keywords : motion estimation, H.264, Video CODEC, circuit architecture

I. Introduction

Motion estimation (ME) achieves a high compression ratio by removing temporal redundancies in the neighboring video sequences. Block-matching motion estimation (BME) is a widely-used technique in many video compression standards such as H.264[1], MPEG-4[2] and VC-1[3]. Since the image

* 정회원, 전자부품연구원 융합신호SoC연구센터
(Convergent SoC Research Center, Korea Electronics Technology Institute)
** 평생회원(교신저자), 한국외국어대학교 전자공학과
(Department of Electronics Engineering, Hankuk University of Foreign Studies)
※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(No.20090067004)과 IDEC의 CAD 툴 지원에 의하여 수행된 연구임.
접수일자: 2010년8월12일, 수정완료일: 2010년11월15일

resolution gets higher and higher, the amount of computation and power consumption required for ME increases dramatically. A variety of BME algorithms have been proposed as a solution. Full search (FS) algorithm guarantees the best results by examining all candidate blocks. However, the usage of FS algorithm is limited because it requires a huge amount of computation and data. Many fast search algorithms such as three-step search[4], four-step search[5], hierarchical search[6] and diamond search[7] have been proposed to reduce the computational efforts. The disadvantage of these algorithms is that data access from or to memory is very irregular and data reuse is difficult. Therefore it is not easy to apply these algorithms in the hardware platform using external memories, e.g., synchronous dynamic random access memory (SDRAM) as frame buffers.

This paper proposes a high-performance integer-pel ME algorithm and circuit architecture that can reduce the bandwidth requirement of external memories. The proposed algorithm consists of coarse and fine searches. In the coarse search, the mean values of 16 pixels in the 4x4 reference block and current block are computed and the FS is performed using those mean values. In addition, four previously determined motion vectors (MV's) are used to improve the quality of the coarse search. In the fine search, the FS is performed for the search range of ±3 around the motion vector (MV) obtained from the coarse search. The irregularity of data access is greatly reduced in the proposed algorithm compared to the conventional hierarchical search. We devised the high-performance circuit architecture for the proposed algorithm and described the circuit at register-transfer level (RTL). The circuit synthesized by using 130nm CMOS standard cell library can process 139.8 1080HD (1,920x1,088) image frames per second and hence support up to H.264 level 5.1.

Ⅱ. Proposed Integer-pel ME Algorithm

In this section, we describe an algorithm to perform the ME efficiently for HD video by reducing the irregularity of the data access. The proposed algorithm consists of two steps - coarse step search and fine step search.

1. Coarse Step ME

We propose an enhanced algorithm to improve the quality of the coarse search as shown in Fig. 1. The mean values of 16 pixels in the 4x4 reference block and current block are computed to construct a coarse block shown as Fig. 2. Since the coarse block area is 1/16 of original one, the amount of data and the number of clock cycles required to compute the sum of absolute differences (SAD) are reduced into 1/16.

Five MV's (MV_{min0} ~ MV_{min4}) are computed using coarse blocks on the completion of (b). MV_{min0} corresponds to the smallest SAD and MV_{min1} corresponds to the next smallest SAD and so on. In our algorithm, we compute MV_{pred}, expected MV of the current macroblock (MB), by using previously determined MV's in the neighbor in order to speed

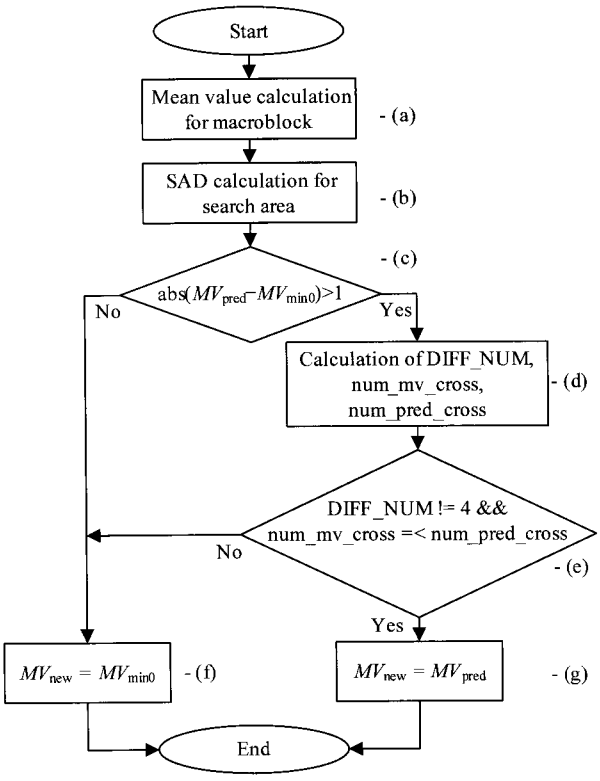


그림 1. Coarse step 움직임 예측 알고리즘
Fig. 1. Algorithm of coarse step motion estimation.

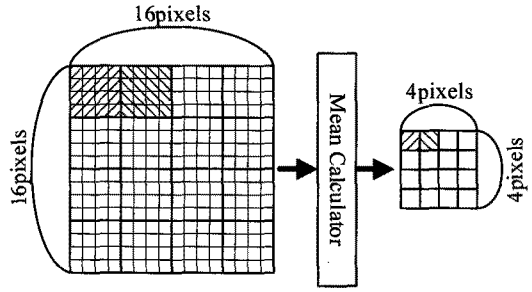


그림 2. 16x16 블록에 대한 평균값 계산
Fig. 2. Mean value calculation for a 16x16 block.

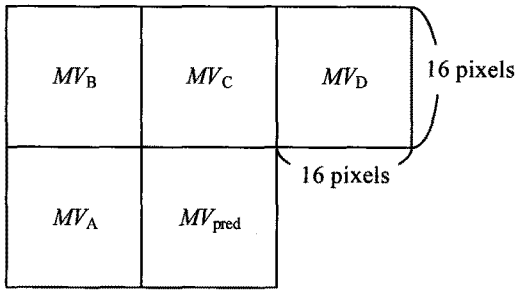


그림 3. 움직임 벡터 예측
Fig. 3. Motion vector prediction.

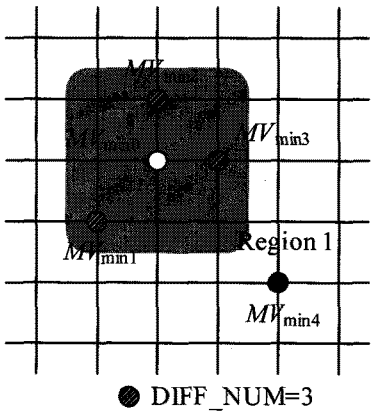


그림 4. DIFF_NUM의 예
Fig. 4. Example of DIFF_NUM.

up the computation process of MV. MV_{pred} is computed by (1) where $MV_A \sim MV_D$ are the previously determined MV's as shown in Fig. 3. If the condition in (c) is not satisfied, the final MV (MV_{new}) is determined as MV_{min0}. Otherwise, the distribution characteristics of MV_{pred} and $MV_{min0} \sim MV_{min4}$ are examined as specified in (d) and (e). DIFF_NUM is the number of MV's in region 1 (center: MV_{min0}) of Fig. 4. num_mv_cross is the number of MV's in region 2 (center: MV_{min0}) and num_pred_cross is the number of MV's in region 3

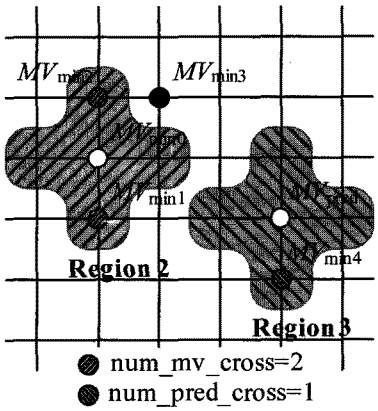


그림 5. num_mv_cross와 num_pred_cross의 예
Fig. 5. Examples of num_mv_cross and num_pred_cross.

(center: MV_{pred}) of Fig. 5. MV_{new} can be either MV_{min0} or MV_{pred} depending on the condition (e).

$$MV_{pred} = (2\ MV_A + MV_B + 2\ MV_C + MV_D + 3)/6 \quad (1)$$

2. Fine Step ME

The MV obtained by the coarse ME has the precision of four pixels and we need to perform one more ME with the precision of one pixel. In the fine ME, the FS is performed for the search range of ±3 around the MV obtained from the coarse ME. Fig. 6 compares the rate distortion (RD) curves of five algorithms: FS, unsymmetrical cross multi-hexagon-grid-search (UMHS), enhanced predictive zonal search (EPZS), proposed algorithm without using MV_{pred} and proposed algorithm with using MV_{pred}. In this figure, FS, EPZS, UMHS, and 'Proposed with MV_{pred}' are indistinguishable because they have almost the same values.

Table 1 shows the results in detail. In the experiments, the proposed algorithms use the coarse step search and then fine step search. The 1080HD test sequence ('Blue sky') consists of one intra (I) frame and 99 predict (P) frames. JM 14.0[8] is used in the experiment and the value of quantization parameter (QP) is set to 5, 15, 28, 38 and 48. The search range is ±16 and the number of reference frame is one. As shown in Fig. 6 and Table 1, the

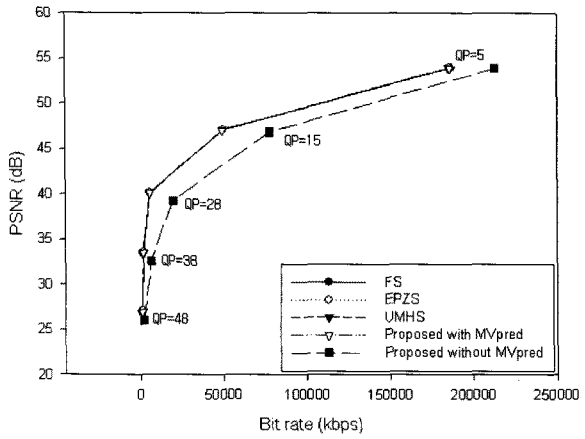


그림 6. 다양한 ME 알고리즘에 대한 'Blue sky'의 RD 곡선

Fig. 6. RD curves of 'Blue sky' for various ME algorithms.

표 1. 다양한 ME 알고리즘의 성능 비교

Table 1. Performance comparison of various ME algorithms..

QP	48	38	28	15	5
△ Bit rate (%)					
FS	0.000	0.000	0.000	0.000	0.000
EPZS	0.025	0.012	-0.004	-0.002	-0.006
UMHS	0.027	-0.042	-0.002	-0.014	-0.009
without MV_{pred}	1.449	2.942	2.672	0.563	0.140
with MV_{pred}	0.082	0.035	0.004	-0.009	-0.007
△ PSNR (dB)					
FS	0.00	0.00	0.00	0.00	0.00
EPZS	0.18	0.07	-0.01	-0.05	-0.02
UMHS	-0.06	-0.10	-0.07	-0.06	-0.03
without MV_{pred}	-0.90	-0.90	-0.87	-0.23	-0.06
with MV_{pred}	0.11	0.03	-0.05	-0.05	-0.02

performance of the proposed algorithm with using MV_{pred} is almost the same as the previous algorithms including FS. In Table 1, the positive bit rate represents the smaller compression ratio compared to FS and the positive PSNR represents the better image quality compared to FS. We obtained the similar results for other test sequences. Fig. 7 compares the bandwidth requirements of external memory for the H.264 encoder. The bandwidth requirements are calculated for one MB in one

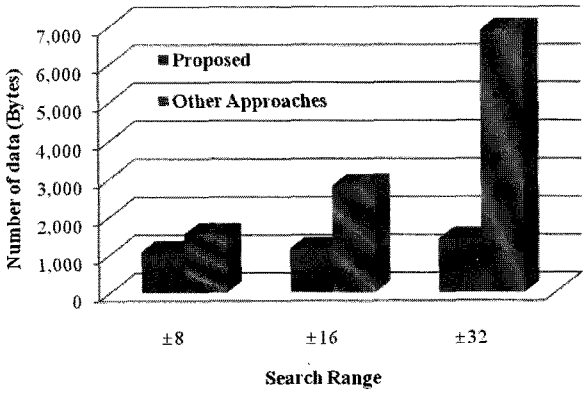


그림 7. 다양한 ME 알고리즘에 대한 요구 대역폭

Fig. 7. Bandwidth requirements for various ME algorithms.

reference frame. In this comparison, 'Other Approaches' include FS and various fast search methods. We assumed that all the data in the search range are read from the external memory in the fast search methods because they require random access of the data. The bandwidth requirement of our algorithm for the search range of ±32 is only 20.04% of others (1,412bytes vs. 6,912bytes).

III. Proposed Circuit Architecture

Fig. 8 shows the architecture of the proposed circuit to implement the coarse and fine step ME's. It consists of SAD Calculator, SAD Comparator, Prediction MV Calculator, current register buffer (CRB), reference register buffer (RRB), Mean Calculator, 14 single-port SRAM's (SPSRAM), SRAM Controller and Motion Estimation Controller. The reference data from external memory are stored in four 22x32-bit SPSRAM's (B_0 ~ B_3) and six 8x32-bit SPSRAM's (B_4 ~ B_9). The data read from SRAM's are stored in RRB shown in Fig. 9. RRB receives four 32-bit inputs (I_0 ~ I_3) simultaneously. Motion Estimation Controller determines which group (Group A, B or C) should be used as inputs of RRB. The reference data move around by ±4 for coarse step ME and ±3 for fine step ME to gather 256 data and transfer them to SAD Calculator. CRB receives data from four 16x32-bit SPSRAM's (B_{10} ~ B_{13}).

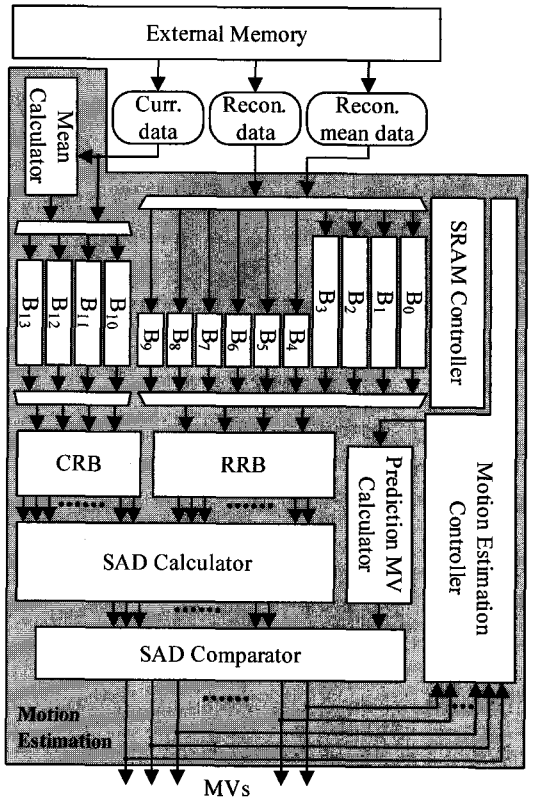


그림 8. 제안한 움직임 예측 회로의 블록도
Fig. 8. Block diagram of proposed motion estimation circuit.

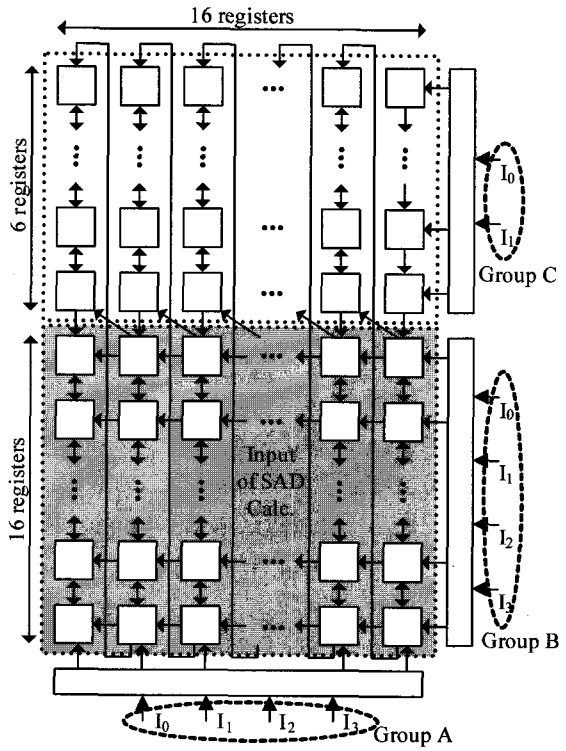
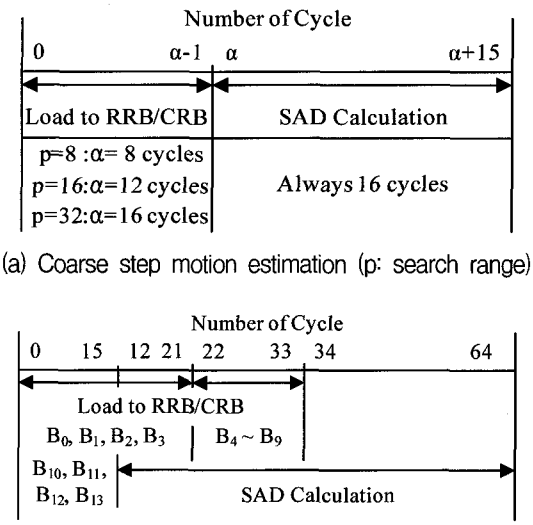


그림 9. Reference 레지스터 버퍼의 블록도
Fig. 9. Block diagram of reference register buffer.



(a) Coarse step motion estimation (p: search range)

(b) Fine step motion estimation

그림 10. 제안한 움직임 예측의 타이밍도
Fig. 10. Timing diagram of proposed motion estimation.

SAD Comparator determines the minimum SAD and 41 MV's. Prediction MV Calculator computes MV_{pred} for coarse step ME. The mean values of current data are computed by Mean Calculator and stored in four buffers (B_{10} ~ B_{13}). The reconstructed data are obtained by the motion compensation and inverse discrete cosine transform (IDCT). The mean values of reconstructed data are computed by Mean Calculator. Both of the reconstructed data and their mean values are stored in the external memory for the ME of the next frame.

Fig. 10 shows the timing diagram of each ME. The coarse step ME requires 12 cycles for receiving data and 16 cycles for SAD computation. The fine step ME requires 16 cycles for storing data in CRB, 34 cycles for storing data in RRB, and 50 cycles for SAD computation. Note that the above discussion assumes that the search range is ±16.

IV. Implementation Results

We designed the circuit at RTL using Verilog. The circuit synthesized by using 130nm CMOS standard cell library consists of 133,153 gates and has the maximum operating frequency of 101.5MHz. Table 2 and Table 3 show the comparison results with other

표 2. 합성 결과 비교
Table 2. Comparison of synthesis results.

	Area (K gates)	On-chip memory (bits)	Max. frequency (MHz)	Process (nm)	Search method
[9]	108	0	100	130	FS
[10]	105	0	100	250	FS
[11]	154	24K	100	180	FS
[12]	597	-	200	180	FS
[13]	71	18K	150	250	FS
[14]	123	0	200	180	FS
[15]	210	62K	260	180	FS
[16]	175	18K	187.7	130	FS
[17]	305	110K	108	180	Fast
[18]	131	64K	66	180	Four step
[19]	180	45K	100	130	Fast
Ours	133	10.7K	101.5	130	Fast

표 3. 다른 구조와의 성능 비교
Table 3. Performance comparison with other approaches.

	# of PE's	Clock cycles (Cycles/MB)	Bandwidth (Kbits/MB)	Max. throughput (fps@1080HD)
[9]	16	4,496	107.90	2.7
[10]	256	296	37.89	41.4
[11]	256	256	32.76	47.9
[12]	256	256	24.57	95.7
[13]	16	4,496	611.46	4.0
[14]	256	256	6.14	95.7
[15]	256	5,216	-	2.9
[16]	31	5,924	-	1.9
[17]	128x8	-	26.8	13.2
[18]	-	-	-	1.5
[19]	-	5,924	134	30.0
Ours	256	89	4.38	139.8

approaches. In these tables, we set the search range to ±8 in order to compare the results in the same condition. As shown in these tables, the number of gates and the maximum operating frequency are comparable to other circuits. Our ME circuit shows an outstanding throughput compared to other approaches. It also requires smaller memory and smaller bandwidth per MB compared to previous fast search methods. If the search range is ±8, our circuit requires 89 cycles per MB and the required bandwidth is 4.38Kbits/MB, which is a significant improvement compared to others. It can process 60 1080HD image frames per second with 45.5MHz clock. If the search range is ±32, our circuit requires 97 cycles per MB and the required bandwidth is 7.07Kbits/MB. The operating frequency required to process 30 CIF (352x288) image frames per second for the search range of ±16 is 1.1MHz.

Ⅳ. 결 론

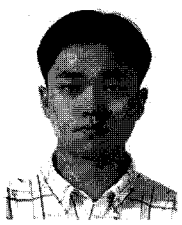
This paper proposes an efficient ME algorithm and the circuit architecture to achieve high speed and low memory bandwidth. By substituting 16 pixels in 4x4 blocks by one mean value during coarse search and considering only the search range of ±3 during fine search, we reduced the amount of data and the number of clock cycles significantly. By using 256 PE's, the resultant circuit processes 139.8 1080HD image frames per second at the operating frequency of 101.5MHz. The proposed method is suitable for the high-quality video compression supporting multiple reference frames and for the low-power video applications such as portable handset devices.

References

[1] Draft IUT-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Mar. 2003.
[2] ISO/IEC 14496-2, Coding of Audio-Visual Objects - part 2: Visual, Nov. 1997.
[3] SMPTE, Standards for Television: VC-1 Compressed Video Bitstream Format and Decoding Process, SMPTE 421M-2006.
[4] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensation interframe coding for video conferencing," Proc. Nat'l Telecomm. Conf., pp.531 - 535, Nov. 1981.
[5] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313 - 317, June 1996.
[6] K. Nam, J. Kim, R. Park, and Y. Shim, "A fast hierarchical motion vector estimation algorithm using mean pyramid," IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 4, pp. 344 - 351, Aug. 1995.

[7] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," IEEE Trans. Image Processing, vol. 9, pp.287 - 290, Feb. 2000.
[8] "H.264/AVC reference software JM14.0", http://iphome.hhi.de/suehring/tml/.
[9] S. Y. Yap and J. V. McCanny, "A VLSI architecture for advanced video coding motion estimation," IEEE Int. Conf. on ASAP, pp.293 - 301, June 2003.
[10] C. Wei and M. Z. Gang, "A novel VLSI architecture for VBSME in MPEG-4 AVC/ H.264," IEEE Int. Sym. on Circuits and Systems, pp.II1794 - 1797, May 2005.
[11] M. Kim, I. Hwang, and S. Chae, "A fast VLSI architecture for full-search variable block size motion estimation in MPEG-4 AVC/H.264," Asia and South Pacific Design Automation Conference, vol. 1, pp. 631 - 634, Jan. 2005.
[12] C. M. Ou, C. F. Le, and W. J. Hwang, "An efficient VLSI architecture for H.264 variable block size motion estimation," IEEE Trans. on Consumer Electronics, vol. 51, pp. 1291 - 1299, Nov. 2005.
[13] C. Wei, M. Z. Gang, L. Z. Qiang, and Z. Yan, "VLSI architecture design for variable-size block motion estimation in MPEG-4 AVC/H.264," IEEE Asia-Pacific Conference on Circuits and Systems, vol. 1, pp. 617 - 620, Dec. 2004.
[14] L. Chen, Y. Zhang, and C. Xu, "Fully utilized and low memory-bandwidth architecture design of variable block-size motion estimation for H.264/AVC," IEEE Region 10 Conference, pp. 1 - 4, Nov. 2006.
[15] L. Deng, W. Gao, M. Z. Hu, and Z. Z. Ji, "An efficient hardware implementation for motion estimation of AVC standard," IEEE Trans. on Consumer Electronics, vol. 51, no. 4, pp. 1360 - 1366, Nov. 2005.
[16] M. Sayed, W. Badawy, and G. Jullien, "Towards an H.264/AVC HW/SW integrated solution: An efficient VBSME architecture," IEEE Trans. on Circuits and Systems-II:Express Briefs, vol. 55, no. 9, pp. 912 - 916, Sep. 2008.
[17] T. Chen, S. Chien, Y. Huang, C. Tsai, C. Chen, T. Chen, and L. Chen, "Analysis and architecture design of and HDTV720p 30 frames/s H.264/AVC encoder," IEEE Trans. on Circuits and Systems for Video Technology, vol. 16, no. 6, pp. 673 - 688, June 2006.
[18] T. Chen, Y. Chen, S. Tsai, S. Chien, and L. Chen, "Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC," IEEE Trans. on Circuits and Systems for Video Technology, vol. 17, no. 5, pp. 568 - 577, May 2007.
[19] C.-C. Lin, Y.-K. Lin, and T.-S. Chang, "PMRME: A parallel multi-resolution motion estimation algorithm and architecture for HDTV sized H.264 video coding," IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. 385 - 388, Apr. 2007.

저 자 소 개



이 선 영(정회원)
1998년 2월 한국외국어대학교 전자공학과 학사 졸업.
2000년 2월 한국외국어대학교 전자공학과 석사 졸업.
2009년 8월 한국외국어대학교 전자공학과 박사 졸업.
2001년 2월～2006년 4월 (주)이시티 반도체연구소 선임 연구원.
2009년 9월～현재 전자부품연구원 융합신호SoC연구센터 선임 연구원.
<주관심분야 : SoC 설계>

조 경 순(평생회원)-교신저자
1982년 2월 서울대학교 전자공학과 학사 졸업.
1984년 2월 서울대학교 전자공학과 석사 졸업.
1988년 12월 미국 Carnegie Mellon University 전기 및 컴퓨터 공학과 박사 졸업.
1988년 11월～1994년 8월 삼성전자(주) 반도체 총괄 선임, 수석 연구원.
1994년 8월～현재 한국외국어대학교 전자공학과 조교수, 부교수, 정교수.
<주관심분야 : SoC 설계>