

Algorithmic and Architectural Co-design for Integer Motion Estimation of AVS

Bin Sheng, Wen Gao, *Member, IEEE*, and Don Xie

Abstract — *The video part of AVS has been finalized. In order to enhance coding performance, AVS video standard adopts some new features for motion estimation, such as variable block size search, multiple reference frames, and motion vector prediction. However, the better performance comes at the price of high computational complexity, data dependence and memory access requirement. These new features also make the hardware implementation more difficult, especially for real-time applications. In this paper, we firstly propose an integer motion estimation algorithm from hardware-oriented viewpoint. Experimental results show that the proposed algorithm has almost the same performance as the reference software of AVS in SDTV applications. Then the corresponding VLSI architecture is presented. The VLSI architecture has been described in Verilog HDL and synthesized using 0.18 μ m Artisan CMOS cells library. The circuit totally costs about 400K equivalent logic gates. At 108MHz working frequency, the circuit can meet the real-time requirement for SDTV (720 \times 576, 25fps) applications, with the search area of 192 \times 192. The co-design obtains better tradeoff between performance and gate-count¹.*

Index Terms — motion estimation, AVS, VLSI, co-design

I. INTRODUCTION

A new audio and video compression standard of China, known as advanced Audio Video coding Standard (AVS), is emerging. This new standard provides a technical solution for many applications within the information industry, such as digital broadcasting, high-density storage media, internet stream media, and so on. The video part of AVS, AVS-P2 [1], mainly targets SDTV and HDTV video compression. Comparing with other international coding standards, such as MPEG-2 [2], MPEG-4 [3] and H.264/AVC [4], the advantages of AVS video standard include higher performance, lower complexity, lower implementation cost and licensing fees.

AVS-P2 adopts the similar block-based hybrid coding framework with MPEG-x and H.26x series of standard. It uses the spatial and temporal predictions to eliminate the spatial and temporal data redundancy, and the prediction errors are transformed, quantized and entropy-encoded. In order to achieve better coding efficiency, AVS-P2 adopts following

new features. The intra prediction defines 5 modes for luma blocks and 4 modes for chroma blocks to make better prediction. To remove more temporal redundancy, the inter prediction adopts variable block sizes (VBS), multiple reference frames (MRF), and quarter-pixel-accurate motion estimation (ME). AVS-P2 defines 5 motion compensation modes for P pictures, which are skip, 16 \times 16, 16 \times 8, 8 \times 16 and 8 \times 8 modes. For B pictures, five different types of motion compensation modes are supported: forward, backward, symmetry-based bi-predictive, skip, and direct [5]. The entropy coding adopts fixed length codes, k-th Exponential-Golomb codes, and context-based 2-D VLC [6]. An 8 \times 8 integer DCT transform named Pre-scaled Integer Transform (PIT) [7], which can be implemented by simple addition and shift operations, is used to eliminate the mismatch during inverse transform. Furthermore, an 8 \times 8-block based adaptive in-loop de-blocking filter is used to reduce the blocking artifacts. Benefiting from these new features, AVS-P2 can achieve more than 50% coding gains over MPEG-2, and similar coding efficiency as H.264/AVC with much lower computational complexity on SDTV and HDTV videos [8].

ME is the most computation-intensive part in AVS-P2 encoder. For SDTV or HDTV real-time coding applications, the acceleration for ME by a dedicated hardware is a prerequisite. The block matching algorithm (BMA) is generally selected for ME in video codec's because of its simplicity and good performance. Among all BMAs, the full-search block matching algorithm (FSBMA) is the most typical one due to its regularity. In the literature, there have been various 1-D and 2-D systolic/semi-systolic array architectures proposed for FSBMA [9-16]. However, for SDTV or HDTV real-time applications, FSBMA demands a huge amount of computational complexity that cannot be easily afforded by hardware accelerator. For example, to cover a search area of 64 \times 64 pixels in NTSC (720 \times 480, 30fps) resolution video, over 300 GIPS (giga-instructions per second) of processing power is required. Therefore, many fast search algorithms have been proposed to reduce the complexity, and many hardware architectures have been developed to meet the real-time constraints. The most representative designs are presented in [17-21]. The trend of architecture design for fast BMAs is toward algorithmic and architectural co-design [22].

Since AVS-P2 is a new standard, few dedicated ME design, which can fully support VBS, MRF and large enough search area for SDTV or HDTV in real time, has been reported by now. Zhang proposed an improved fast FSBMA and VLSI architecture for AVS-P2 in [23]. Working at 150MHz and costing 4257 cycles to process one macroblock (MB) in 65 \times 65

¹ This work was supported by the National High Technology Development 863 Program of China under Grant No. 2003AA1Z1290.

Bin Sheng is with the Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin, China (e-mail: bsheng@jdl.ac.cn).

Wen Gao is with the School of Electronics Engineering and Computer Science, Peking University, China (e-mail: wgao@jdl.ac.cn).

Don Xie is with the Institute of Computing Technology, Chinese Academy of Sciences, China (e-mail: xdxie@jdl.ac.cn).

search area, his architecture can not meet the real-time requirement for SDTV applications. In this paper, we propose a hardware-oriented integer ME (IME) algorithm and its VLSI architecture for AVS-P2 real-time SDTV applications. The proposed algorithm is based on hierarchical motion estimation (HME) algorithm. Combined with motion tracing from neighbor frames, the proposed algorithm can achieve much larger search range with less computational load than traditional HME.

The following sections are organized as follows. Section II presents the hardware-oriented IME algorithm. The VLSI architecture is described in Section III. Section IV presents experimental results and VLSI implementation. Conclusion is given in the last section.

II. HARDWARE-ORIENTED IME ALGORITHM

ME is a procedure to find the motion vector (MV) that represents the spatial displacement of a template block in the current picture from a predicted block in a reference picture. IME is the ME on integer-pixel precision pictures. Compared with IME, the fractional ME (FME) denotes the ME on fractional-pixel precision pictures, such as half-pixel ME and quarter-pixel ME. Since the search area of FME is 3×3 commonly, the computational load of IME is much heavier than that of FME. Although FSBMA can achieve precise IME and can detect the best MV, it requires too much processing power when search area is large. Therefore, finding a fast IME algorithm that can reduce the heavy computation of FSBMA with acceptable video quality degradation is necessary. On the other hand, IME algorithm of RM52C [24] adopts many sequential processes to enhance the coding performance. It is hard to efficiently map the sequential algorithms to parallel hardware architecture. So we adopt some hardware-oriented modifications at first.

A. Hardware-oriented modifications

During ME processing, reference pictures of the current one are commonly those previously reconstructed ones. For example, I0B1B2P3B4B5P6 is an original input sequence in Fig.1, where I denotes Intra picture, P denotes Predicted picture, and B denotes Bidirectionally-predicted picture. The encoding order is I0P3B1B2P6B4B5. In the encoding procedure, P3 refers to I0's reconstructed picture, B1 and B2 refer to the reconstructed pictures of I0 or P3, P6 refers to the reconstructed pictures of I0 or P3 (AVS-P2 can support at most two reference pictures), B4 and B5 refer to the reconstructed pictures of P3 or P6.

In order to increase the degree of concurrence, we use not the reconstructed pictures but the original pictures as reference during ME processing. This replacement is not accurate and will affect the results of ME. But the results are only used in the inter Lagrangian mode selection, where MV cost is calculated for each mode of one macroblock. The processes after the mode selection are still based on the reconstructed pictures. Therefore, the replacement can only affects the

performance of mode selection, and the output of encoder is still compatible with AVS-P2 standard. According to the experimental results, the replacement has little bad impact on the coding performance. Whereas, benefited from this modification, the ME processing order can be the same as the sequence input order. For example, we can start the original-picture-based ME for B1 and B2 before their backward reference P3 is processed. According to the proposed IME algorithm, one MB in P3 can use scaled MV of the corresponding MB in B2 as the search center during its IME. This modification can greatly increase the computational concurrence of IME between neighbor pictures. Moreover, taking the advantage of motion tracing, the modification can help to achieve much better search results for sequences with large motion distance.

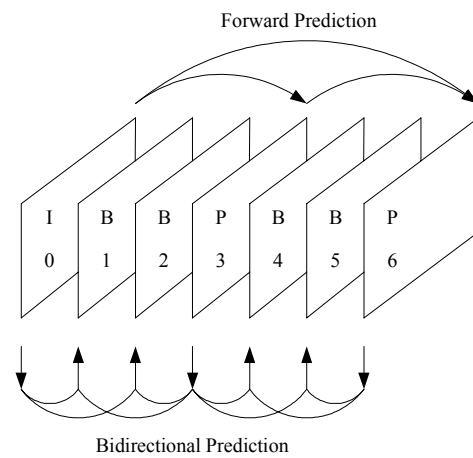
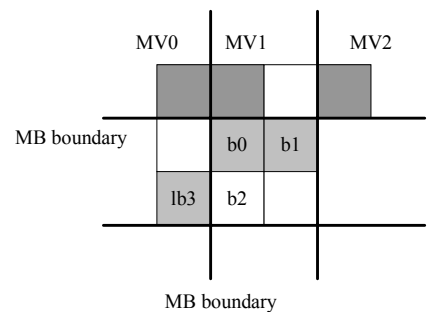


Fig. 1. Input sequence order and the modified ME processing order



Exact MVP of b2 = medium(MVb0, MVb1, MVlb3)
 Modified MVP of b2 = medium(MV0, MV1, MV2)

Fig.2. Modification of MVP

Generally, the MV of each block is predicated by the medium values of MVs from left, top and top-right neighboring blocks. If top-right block is not available, top-left block is used in stead. Therefore, the Lagrangian cost can only be computed after the MVs of the neighboring blocks being determined, which inevitably caused a sequential processing. That is, blocks in an MB and MBs can not be processed in parallel. To solve this problem, the exact MVPs (MV

Predictor) of all nine blocks of an MB, which are the medium of MVs of the left, top, and top-right blocks, are changed to the medium of MVs of the top-left, top, and top-right MBs, as shown in Fig.2.

B. The proposed IME algorithm

The proposed algorithm adopts a three-step hierarchical search scheme, which employs sub-sampling to reduce the computational load greatly. Level 1 is the 4-pixel precision picture, which is a 16-to-1 sub-sampling from the original picture. Level 2 is the 2-pixel precision picture, which is a 4-to-1 sub-sampling from the original picture. Level 3 is the full-pixel precision, i.e. original picture. At each level, FSBMA is adopted to guarantee the highest search performance. The regular search pattern of FSBMA is suitable for parallel processing. Furthermore, FSBMA can effectively support VBS-ME by reusing smaller blocks' SADs (Sum of Absolute Difference) for larger blocks.

The simplified Rate-Distortion cost function is used for MV trimming, in which the best candidates of each mode are evaluated by their Rate-Distortion costs. The weighted SAD is defined in (1).

$$SAD_{weighted} = \alpha \cdot SAD_{mode} + \beta \cdot MVD_{sum} \quad (1)$$

Here, α and β are weighted factors, " SAD_{mode} " denotes the minimal SAD of an MB for each mode of 16×16 , 16×8 , 8×16 and 8×8 . " MVD_{sum} " is defined as

$$MVD_{sum} = \sum_{MVs} (|MV_x - MVP_x| + |MV_y - MVP_y|) \quad (2)$$

" MVs " in equation (2) represents the obtained MVs corresponding to each mode.

For the first B-picture after a reference picture, such as B1 and B4 in Fig. 1, the forward ME processing is described as follows.

Step 1: Full search at Level 1 – The search range is $[-23, 24]$; the search center is $(0, 0)$, which is the up-left pixel of the corresponding block in the reference picture; and the template block size is 4×4 – The outputs are the best two candidate MVs, named C1 and C2, then go to Step2.

Step 2: Full search at Level 2 – The search range is $[-2, 2]$; the search centers are C1, C2, and $(0, 0)$ respectively; the template block size is 8×8 – If any SAD is less than the threshold TH1, the output are the best two candidate MVs, named C3 and C4, then go to Step 4; otherwise go to Step 3.

Step 3: Full search at Level 2 – The search range is $[-15, 16]$; the search center is $(0, 0)$; and the template block size is 8×8 – Select the best two modes from 16×16 , 16×8 , 8×16 , and 8×8 modes, and the outputs are the best MVs for the two selected modes, then go to Step 5.

Step 4: Full search at Level 3 – The search range is $[-2, 2]$; the search centers are C3, C4, and $(0, 0)$ respectively; the

template block size is 16×16 – The output is the best MV, and the block mode is set as 16×16 , then go to Step 6.

Step 5: Full search at Level 3 – The search range is $[-2, 2]$; the search centers are the outputs from Step 3 and $(0, 0)$; the block size is 16×16 – Four sets of MVs will be selected, which are the best MVs for the four block modes respectively, then go to Step 6.

Step 6: The search is over.

For the second B-picture after a reference picture, such as B2 and B5 in Fig. 1, the forward ME processing is described as follows.

Step 0: Full search at Level 3 – The search range is $[-4, 4]$; the search centers are $(0, 0)$ and the scaled MVs of the corresponding MB in the previous B-picture, as illustrated in Fig. 3; the template block size is 16×16 – If any SAD is less than the threshold TH2, the best MV will be output and the mode is selected as 16×16 , then go to Step 6; otherwise, go to Step 1.

Step 1-6 are the same as those of the first B-picture, which have been mentioned above.

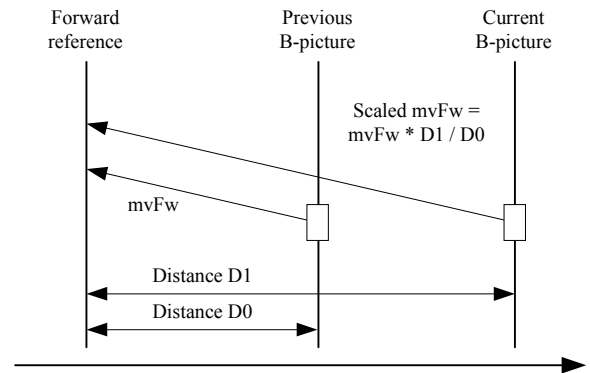


Fig. 3. Scaled MV of corresponding MB in the previous B-picture

For backward ME of B-pictures, the processing is very similar. The only difference is the reversed direction. The B-picture before a reference picture, such as B2 and B5 in Fig.1, is firstly processed. Then the next B-picture, such as B1 and B4 in Fig. 1, is processed using the scaled MVs as the search centers in succession.

P-pictures may have two reference pictures according to AVS-P2. For the nearer forward reference picture, Step 0 to 6 are used. If the farther forward reference picture exists, Step 1 to 6 will be employed.

III. THE PROPOSED ARCHITECTURE

The IME accelerator is designed as a co-processor of an embedded RISC processor. The accelerator adopts the typical two-bus architecture. The RISC processor can communicate with functional modules on the command bus via the co-processor interface. And functional modules can access external DRAM via the data bus. Sub-sampling filter (SSF) takes charge of 16-to-1 or 4-to-1 sub-sampling from the original pictures. Search engine at Level 1 (SE-L1) can

provide FSBMA for 4×4-block in 4-pixel precision pictures. The largest search range of SE-L1 is [-23, 24] in both horizontal and vertical directions. SE-L2 can perform FSBMA for 8×8-block in 2-pixel precision pictures. Its search range can be up to [-15, 16]. Furthermore, SE-L2 supports VBS-ME with a merging scheme. That is, all nine SADs of 8×8, 8×4, 4×8, and 4×4 blocks can be sent out together. SE-L3 can accomplish FSBMA for 16×16-block in full-pixel precision. Its search range is [-4, 4]. Similar to SE-L2, VBS-BMA is supported by SE-L3. It can work out all nine SADs of 16×16, 16×8, 8×16, and 8×8 blocks at the same time. One set of search engines (SE-L1, SE-L2, and SE-L3) can perform hierarchical ME for one reference picture. In order to support MRF-ME in real-time, we utilize two sets of search engines, named SET-1 and SET-2. When processing B-pictures, one set performs forward reference IME, and the other set performs backward reference IME. When processing P-pictures, each set undertakes the IME on one reference picture. Two sets are enough for AVS-P2, which supports at most two reference pictures.

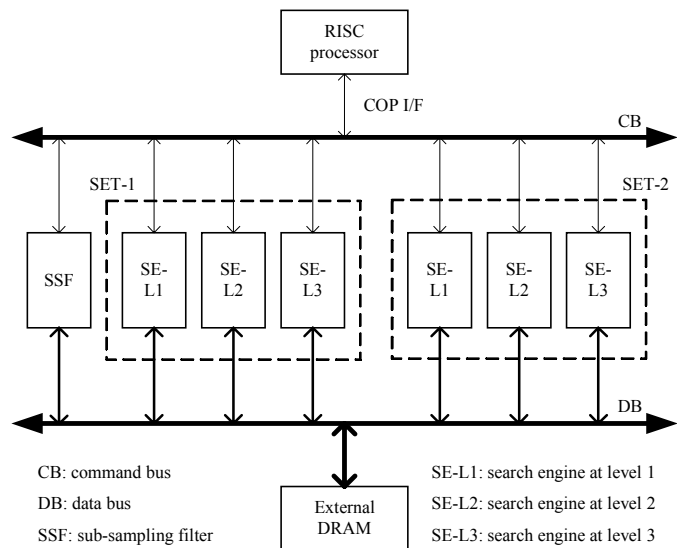


Fig. 4. Block diagram of the IME accelerator

A. Control scheme for the embedded RISC processor

The RISC processor can control functional modules of the accelerator via the co-processor interface, as described in Fig.5. The operations of functional modules are fully parameterized. All registers for parameters and results in functional modules are mapped into the address space of the RISC processor. In order to support MB-level pipeline, the registers for parameters and results are all double-buffered. When a functional module is using the parameters for current operation, software can configure the ones for the next operation.

B. Architecture of the search engine

Architectures of SE-L1, SE-L2, and SE-L3 are very similar. They all have process element (PE) array, adder tree, selection unit, and et al. We select SE-L2 as an example to describe the

architecture of the search engine. SE-L2 supports 8×8-block matching with the search range of [-15, 16]. Furthermore, it can calculate out all nine SADs for 8×8, 8×4, 4×8, and 4×4 blocks to support VBS. Fig. 6 shows the block diagram of SE-L2.

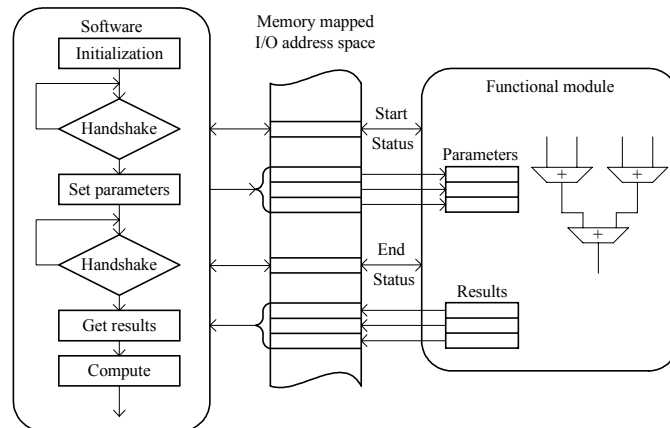


Fig. 5. Interface between RISC processor and functional modules

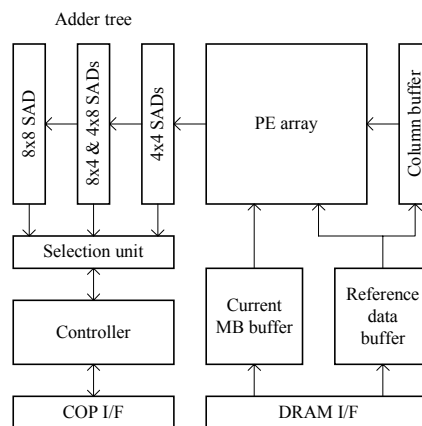


Fig. 6. Block diagram of SE-L2

Current MB buffer and reference data buffer are two on-chip SRAMs, which are used to store pixel data for block matching. PE array has 8×8 PEs corresponding to 8×8 block size. Each PE calculates one absolute difference (AD). The PE array can calculate 64 ADs in parallel. The 64 ADs are then sent into the adder tree to generate nine SADs for various block sizes. In selection unit, the best matching mode and MV are decided and sent back to RISC processor via COP interface.

The macro architecture of an N×N PE array is shown in Fig. 7. C and R are two 8-bit registers used to store a template pixel and a reference pixel respectively. AD is another 8-bit register for storing absolute difference between C and R. The value of register C can be transferred to that of the upper PE. And the value of register R can be passed to the upper, the lower, or the left PE. Assuming the search range is M×M and all pixels are ready in the buffers, the array architecture needs M²+N cycles to finish the full search.

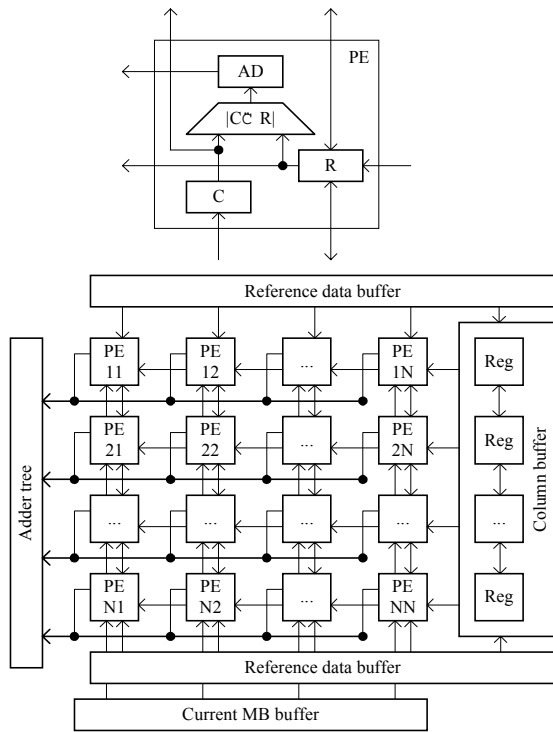


Fig. 7. Marco architecture of N×N PE array

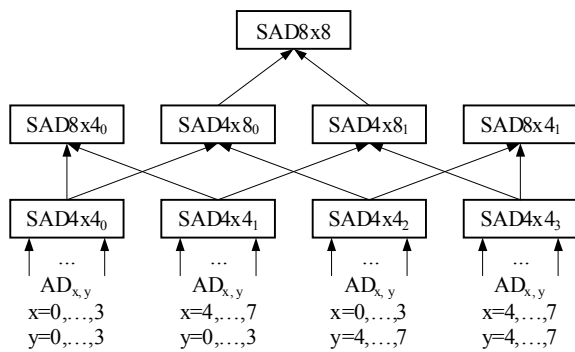


Fig. 8. The merging scheme for 8×8 block in SE-L2

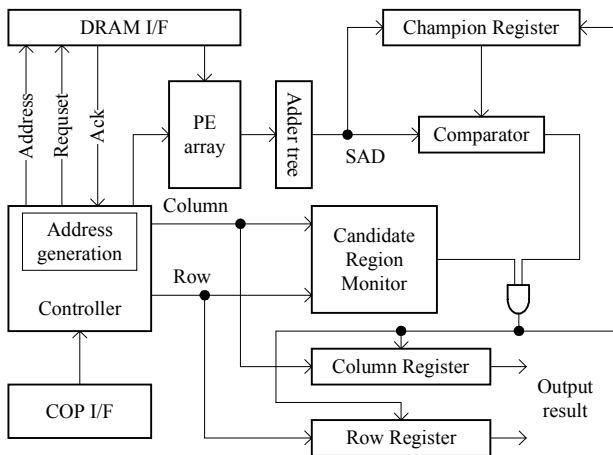


Fig. 9. Marco architecture of the selection unit

The ADs are sent out from PE array to the adder tree in parallel. The adder tree has a three-stage pipeline, which can implement the merging scheme described in Fig. 8.

Fig. 9 illustrates the macro architecture of the selection unit. The champion register always stores the minimal SAD. Candidate region monitor is used to guarantee that the values of column and row are in valid range.

In summary, the proposed architecture takes full advantage of the concurrence provided by the proposed hardware-oriented IME algorithm. SET-1 and SET-2 can work for two pictures concurrently. Since there is no dependence between neighbor MBs, the pipeline consisted of SE-L1, SE-L2, and SE-L3 can process MB by MB at three resolution levels. Among the three search engines, SE-L1 costs the most cycles, i.e. about 2308, to process one MB because its search area is the largest. In the worst case, about 2500 cycles are needed for the three-stage pipeline to process one MB, including extra operations such as buffer loading.

IV. EXPERIMENTAL RESULTS

To evaluate the performance, we have tested the proposed algorithm using three standard test sequences, which are “basketball”, “flower-garden” and “horse-riding”. Each sequence contains 150 frames. The coding sequence is selected as “IPBB”. Weighted prediction, rate control, and de-blocking filter are all disabled. Two reference frames, 192×192 search area, and all block sizes are used. Five QPs are selected, which are 28, 32, 36, 40, and 44. The comparisons of PSNR and bit rate between the proposed algorithm and RM52C, the reference software of AVS-P2, are illustrated in Fig. 10, 11, and 12 respectively. From the curves, we can see that the coding efficiency of the proposed algorithm is nearly the same as that of RM52C.

The proposed architecture is described in Verilog HDL at register transfer level (RTL), and synthesized using 0.18μm Artisan CMOS cells library by Synopsys Design Compiler. One set of search engines costs about 198K equivalent logic gates. The whole circuit, including two sets of search engines and sub-sampling filter, costs about 400K equivalent logic gates. The critical path is less than 6ns in the worst case. Table I shows the comparison of synthesized results between Zhang’s design in [23] and ours.

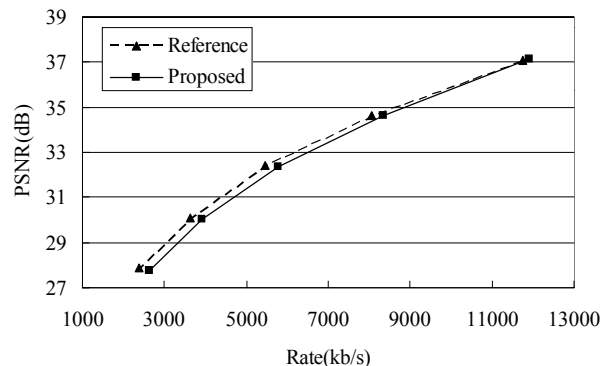


Fig. 10. Rate-Distortion curves of “basketball”

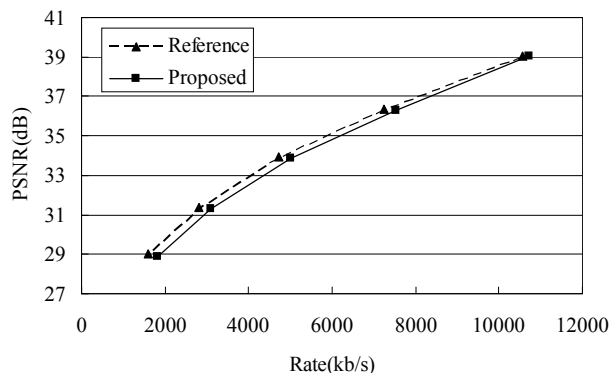


Fig. 11. Rate-Distortion curves of "flower-garden"

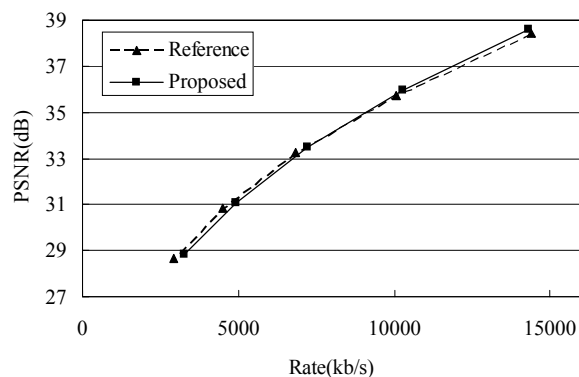


Fig. 12. Rate-Distortion curves of "horse-riding"

TABLE I
COMPARISON OF SYNTHESIZED RESULTS

	Zhang's Architecture	Our Architecture
Search area	65 × 65	192 × 192
Cycle per MB	4,257	2,500
Technology	0.18μm Artisan CMOS	0.18μm Artisan CMOS
Working frequency	150MHz	108MHz
Gate count	212K (1 reference frame)	400K (2 reference frames)
Capacity	Can not support SDTV	SDTV (720×576, 25fps)

V. CONCLUSION

In this paper, we present an algorithmic and architecture co-design for motion estimation of AVS-P2. The proposed algorithm combines the hierarchical ME with the motion tracing from neighbor frames. Moreover, some hardware-oriented modifications are adopted in the proposed algorithm to increase the concurrence. Experimental results show that the proposed algorithm has almost the same performance as the reference software RM52C of AVS-P2 in SDTV encoding applications. The corresponding VLSI architecture is also proposed. As a co-processor of an embedded RISC processor, the proposed architecture can perform integer motion estimation on two reference frames and support variable block

size. The synthesis results show that the architecture can support real-time integer motion estimation for AVS-P2 SDTV sequences (720×576, 25fps) with 192×192 search area at 108MHz working frequency.

REFERENCES

- [1] Audio Video Coding Standard Workgroup of China (AVS), Advanced Coding of Audio and Video - Part 2: Video, December 2004.
- [2] ISO/IEC IS 13818, General Coding of Moving Picture and Associated Audio Information, 1994.
- [3] Information technology - Coding of audio-visual objects - Part 2: Visual (ISO/IEC FCD 14496), July 2001.
- [4] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), May 2003
- [5] X.Y. Ji, D.B. Zhao. New bi-prediction techniques for B pictures coding. *Proceedings of 2004 IEEE International Conference on Multimedia and Expo (ICME2004)*, Taipei, June 2004, 1: 101-104.
- [6] Q. Wang, D.B. Zhao, S.W. Ma. Context-based 2D-VLC for video coding. *Proceedings of 2004 IEEE International Conference on Multimedia and Expo (ICME2004)*, Taipei, June 2004, 1: 89-92.
- [7] C.X. Zhang, J. Lou, L. Yu. The technique of pre-scaled integer transform. *Proceedings of 2005 IEEE International Symposium on Circuits and Systems (ISCAS2005)*, Kobe, Japan, May 2005, 1: 316-319.
- [8] L. Yu, F. Yi, J. Dong, C.X. Zhang. *Proceedings of 2005 Visual Communications and Image Processing (VCIP2005)*, 2005, 679-690.
- [9] T. Komarek, P. Pirsch. Array architectures for block matching algorithms. *IEEE Transactions on Circuits and Systems*, 1989, 36 (10): 1301-1308.
- [10] L.D. Vos, M. Stegherr. Parameterizable VLSI architectures for the full-search block-matching algorithm. *IEEE Transactions on Circuits and Systems*, 1989, 36(10):1309-1316.
- [11] K.M. Yang, M.T. Sun, L. Wu. A family of VLSI designs for the motion compensation block-matching algorithm. *IEEE Transactions on Circuits and Systems*, 1989, 36(10): 1317-1325.
- [12] C.H. Chou, Y.C. Chen. A VLSI architecture for real-time and flexible image template matching. *IEEE Transactions on Circuits and Systems*, 1989, 36 (10):1336-1342.
- [13] C.H. Hsieh, T.P. Lin. VLSI architecture for block-matching motion estimation algorithm. *IEEE Transactions on Circuits and Systems for Video Technology*, 1992, 2(2):169-175.
- [14] H. Yeo, Y.H. Hu. A novel modular systolic array architecture for full-search block matching motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1995, 5(5):407-416
- [15] Y.K. Lai, L.G. Chen. A data-interlacing architecture with two-dimensional data-reuse for full-search block-matching algorithm. *IEEE Transactions on Circuits and Systems for Video Technology*, 1998, 8(2):124-127.
- [16] Y.H. Yeh, C.Y. Lee. Cost-effective VLSI architectures and buffer size optimization for full-search block matching algorithms. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 1999, 7(3):345-358.
- [17] H. M. Jong, L. G. Chen, T. D. Chiueh. Parallel architectures for 3-step hierarchical search block-matching algorithm. *IEEE Transactions on Circuits and Systems for Video Technology*, 1994, 4(4):407-416
- [18] S. Futta, W. Wolf. A flexible parallel architecture adopted to block-matching motion estimation algorithms. *IEEE Transactions on Circuits and Systems for Video Technology*, 1996, 6(1):74-86
- [19] M. Mizuno, Y. Ooi, N. Hayashi, et al. A 1.5-W single-chip MPEG-2MP@ML video encoder with low power motion estimation and clocking. *IEEE Journal of Solid-State Circuits*, 1997, 32(11):1807-1816.
- [20] M. Takahashi, T. Nishikawa, M. Hamada, et al. A 60-mW MPEG4 video codec using clustered voltage scaling with variable supply-voltage scheme. *IEEE Journal of Solid-State Circuits*, 1998, 33(11):1772-1780.
- [21] J. H. Lee, K. W. Lim, B. C. Song, J. B. Ra. A fast multi-resolution block matching algorithm and its VLSI architecture for low bit-rate video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2001, 11(12):1289-1301.

- [22] T. C. Chen, S. Y. Chien, et al. Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder. *IEEE Transactions on Circuits and Systems for Video Technology*, 2006, 16(6):673-688
- [23] L. Zhang, D. Xie and D. Wu. Improved FFSSBM algorithm and its VLSI architecture for AVS video standard. *Journal of Computer Science and Technology*, 2006, 21(3):378-382.
- [24] AVS 1.0 RM52C, December 2004.



Bin Sheng is a Ph.D. candidate at Harbin Institute of Technology. He received his MS degree in Computer Science from Harbin Institute of Technology, China, 2001. His research interests include: VLSI design, image processing, multimedia data compression.



Wen Gao (M'99) received the M.S. degree and the Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1985 and 1988, respectively, and the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991. He was a Research Fellow with the Institute of Medical Electronics Engineering, University of Tokyo, in 1992, and a Visiting Professor with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, in 1993. From 1994 to 1995, he was a Visiting Professor with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge. Currently, he is the Director of the Joint R&D Lab (JDL) for Advanced Computing and Communication, Chinese Academy of Sciences, a Professor with the School of Electronics Engineering and Computer Science, Peking University, a Professor of computer science with the Harbin Institute of Technology, and an honor Professor of computer science at City University of Hong Kong. He has published seven books and over 200 scientific papers. His research interests are in the areas of signal processing, image and video communication, computer vision, and artificial intelligence. Dr. Gao chairs the Audio Video coding Standard (AVS) workgroup of China. He is the head of the Chinese National Delegation to MPEG working group (ISO/SC29/WG11). He is also the Editor-in-Chief of the Chinese Journal of Computers and the general Co-Chair of the IEEE International Conference on Multi-model Interface in 2002.



Don Xie received his Ph.D. degree in Electrical Engineering, University of Rochester, USA. He was a Senior Scientist at Eastman Kodak Company, New York, USA, from 1994 to 1997; a Principal Scientist at Broadcom Corporation, California, USA, from 1997 to 2002. Now he is a Researcher of Institute of Computing Technology, Chinese Academy of Sciences. His research interests include multimedia SoC design, embedded system. He has 23 U.S. Patents.