

Fast Algorithm and Architecture Design of Low-Power Integer Motion Estimation for H.264/AVC

Tung-Chien Chen, Yu-Han Chen, Sung-Fang Tsai, Shao-Yi Chien, and Liang-Gee Chen, *Fellow, IEEE*

Abstract—In an H.264/AVC video encoder, integer motion estimation (IME) requires 74.29% computational complexity and 77.49% memory access and becomes the most critical component for low-power applications. According to our analysis, an optimal low-power IME engine should be a parallel hardware architecture supporting fast algorithms and efficient data reuse (DR). In this paper, a hardware-oriented fast algorithm is proposed with the intra-/inter-candidate DR considerations. In addition, based on the systolic array and 2-D adder tree architecture, a ladder-shaped search window data arrangement and an advanced searching flow are proposed to efficiently support inter-candidate DR and reduce latency cycles. According to the implementation results, 97% computational complexity is saved by the proposed fast algorithm. In addition, 77.6% memory bandwidth is further saved with the proposed DR techniques at architecture level. In the ultra-low-power mode, the power consumption is 2.13 mW for real-time encoding CIF 30-fps videos at 13.5-MHz operating frequency.

Index Terms—ITU-T Rec. H.264, ISO/IEC 14496-10 AVC, motion estimation (ME), VLSI architecture.

I. INTRODUCTION

H.264/AVC [1] can save 25%–45% and 50%–70% of bitrates compared with MPEG-4 Advanced Simple Profile (ASP) and MPEG-2, respectively [2]. Many new features [3]–[5] are used to achieve much better rate-distortion efficiency and subjective quality, but the high computational complexity is the penalty. According to the instruction profile, an H.264/AVC encoder requires 315 Giga-instructions per second (GIPS) computation and 471 Giga-bytes per second (GByte/s) memory access to encode a CIF 30-fps video [6]. Such high requirement of computational resources leads to high power consumption. For portable and wearable devices, in which the power resource is limited, low-power design techniques are essential.

For a low-power H.264/AVC video encoder, the most critical component should be integer motion estimation (IME). The IME requires 74.29% (234 GIPS) computation and 77.49% (365 GByte/s) memory access requirement of the whole encoder [6]. Compared with the previous standards, the IME of H.264/AVC

is almost ten times more complex than that in MPEG-4 [6], [7]. This is caused by the new prediction tools of variable block sizes (VBS) and multiple reference frames (MRF).

In the IME algorithm, the current frame is partitioned into many macroblocks (MBs). For each current MB (CMB) in the current frame, one best matched block which is the most similar to this current MB is looked for within a search window (SW) of reference frame. The IME calculates the matching costs of candidates in SW, and the candidate with the smallest matching cost is the best match. The most common criterion of the matching cost is the sum of absolute differences (SADs) between current pixels of CMB and reference pixels of each candidate.

In a typical IME module, reference pixels of the SW are stored in local memories, and matching costs are calculated by parallel processing elements. The power consumption of the IME module mainly comes from two parts. The first one is the data access power to read reference pixels from local memories. The other is computational power to calculate matching costs with processing elements. Several techniques are used to reduce the power consumption. At the architecture level, because the reference pixels of neighboring candidates are considerably overlapped, the reference pixels read from local memories are stored in registers and reused by parallel processing elements. This is called the candidate-level data reuse (DR), and the data access power is reduced. At the algorithm level, fast algorithms are applied to reduce the computational complexity. Both the data access power and the computational power are thus saved.

For previous H.264/AVC IME designs, several hardware architectures were proposed to support a full search (FS), i.e., exhausted search, algorithm [8]–[12]. They provide good candidate-level DR with regular searching flows, but the computational complexity is large because of the exhausted search. On the other hand, for the previous standards, several low-power IME architectures [13]–[15] with corresponding fast algorithms were designed. However, the functionalities of H.264/AVC are not supported. In addition, because the irregular searching flows of fast algorithms usually lead to poor inter-candidate DR, the power reduction at the algorithm level usually forms constraints for the power reduction at architecture level. Therefore, a new low-power IME architecture is urgently demanded for H.264/AVC encoders. Some advanced techniques are required to efficiently combine the inter-candidate DR with fast algorithms.

In this paper, a fast algorithm with several hardware considerations is proposed to support H.264/AVC IME. In addition, a parallel architecture is designed to support this fast algorithm with efficient inter-candidate DR. The remainder of this

Manuscript received March 25, 2006; revised August 21, 2006. This work was supported in part by the National Science Council, Taiwan, R.O.C., under Grant 95PFA0106257. This paper was recommended by Associate Editor C. N. Taylor.

The authors are with the DSP/IC Design Laboratory, Department of Electrical Engineering and Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: djchen@video.ee.ntu.edu.tw; doliamo@video.ee.ntu.edu.tw; bigmac@video.ee.ntu.edu.tw; sychien@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2007.894044

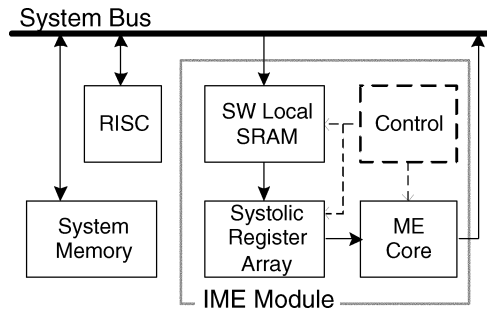


Fig. 1. Block diagram of the IME system architecture.

paper is organized as follows. In Section II, the power reduction techniques are reviewed followed by problem definitions. In Section III, a hardware-oriented fast algorithm is proposed with the consideration of candidate-level DR. In Section IV, the corresponding architecture is designed with similar DR capability compared with FS IME architectures. The implementation results and comparisons are shown in Section V. Finally, Section VI presents the conclusion.

II. FUNDAMENTAL AND PROBLEM DEFINITION

A. Power Reduction Techniques

Fig. 1 shows the typical hardware architecture of IME module. Three techniques are investigated to reduce the power consumption. The first technique is the MB-level DR. Because SWs of neighboring CMBs are considerably overlapped, the SW SRAMs are generally embedded as the cache memories. The reference pixels read from system memory can be stored and reused locally in the SW SRAMs in the IME module. The power consumption of system memory and system bus is thus saved. The second one is fast algorithms. This technique can reduce the searched candidate number or referred pixel number of each candidate. It can save both the computational power of the ME core and the data access power of the SW SRAMs. As for the third technique, because pixels of neighboring candidates are also overlapped, systolic register arrays with corresponding parallel ME core are designed to achieve the candidate-level DR. The reference pixels read from the SW SRAMs are shifted in the systolic array and reused by the ME core. The data access power of the SW SRAMs is further reduced with an additional power consumption of systolic register array. It is worth it because SRAMs usually consume much more power than register circuits.

For MB-level DR, four DR schemes indexed from level A to level D have been proposed with different tradeoffs between local memory size and system bus bandwidth [16]. Level A requires the smallest local memory size and the highest external bandwidth, while level D has the largest local memory size and the lowest external bandwidth. Furthermore, H.264/AVC supports multiple-reference-frame ME (MRF-ME), and the required system bandwidth is increased in proportion to the reference frame number. A single-reference-frame multiple current MB (SRMC) scheme has been proposed to further exploit the DR at the frame level [17]. These schemes are used to reduce the power consumption outside the IME module and are orthogonal to fast algorithms and candidate-level DR schemes. In this

paper, we will focus on the low-power techniques within the IME module.

B. Problem Statements

The candidate-level DR is very important for low-power IME module. A key factor is to efficiently combine IME algorithms and parallel hardware architectures. In the following, the concepts of candidate-level DR will first be described based on the FS (exhausted search) algorithm. Two categories of candidate-level DR schemes will be introduced. Then, we will state the cooperative problems between fast algorithms and parallel hardware in terms of candidate-level DR.

In parallel architectures, two kinds of candidate-level DR schemes are generally used with the FS algorithm. First, all distortion costs (SADs) of the smallest 4×4 blocks are computed first. The costs of larger block sizes are calculated online by summing up the corresponding 4×4 costs [9]–[11], [18]. This reuse scheme is called intra-candidate DR. Furthermore, the search pattern to support the FS algorithm is regular. The reference pixels can be easily reused by neighboring candidates [9]–[11], which is called inter-candidate DR scheme.

Traditional fast algorithms such as three step search (3SS) [19], four step search (4SS) [20], and diamond search (DS) [21] are developed for fixed block size. They cannot efficiently support variable block size ME (VBS-ME) for H.264/AVC. For VBS-ME, the matching costs of 41 blocks may saturate in different directions. In order to maintain the performance of VBS-ME, the searching algorithm is repeated 41 times for different block sizes. Because the variable blocks can form seven 16×16 blocks, approximately seven times the computational complexity is required compared with the previous standards. In addition, the hardware architecture for these fast algorithms [13]–[15] can not support inter-candidate DR as efficiently as the architectures for the FS algorithm. The candidates in 3SS are far from each other. The pattern with diagonal direction in DS make the inter-candidate DR inefficient. In addition, the irregular and sequential searching path in DS and FSS also lead to a poorer DR rate, which will be described more in Section IV-A.

Several new fast algorithms for VBS-ME have been proposed in recent years. In [22], Chan *et al.* proposed a top-down procedure to process the largest 16×16 block first. Then, the remaining blocks are processed if needed. In [23], a bottom-up approach starting from the smallest 4×4 blocks was suggested by Rhee *et al.* By combining the above two ideas, Zhou *et al.* proposed a merge-and-split scheme in [24]. These algorithms are all performed sequentially with predefined criteria, and the computation can be reduced by the early termination. However, for hardware implementation, the irregular flows result in complex control circuits. The sequential procedures of variable blocks restrict the intra-candidate DR scheme.

In summary, a new parallel IME architecture with hardware-oriented fast algorithm is urgently needed in H.264/AVC systems for portable devices. The fast algorithm should not only reduce the computational complexity but also consider the DR capability for hardware implementation. In addition, advanced techniques at the architecture level should also be utilized to enable the parallel processing for sequential and

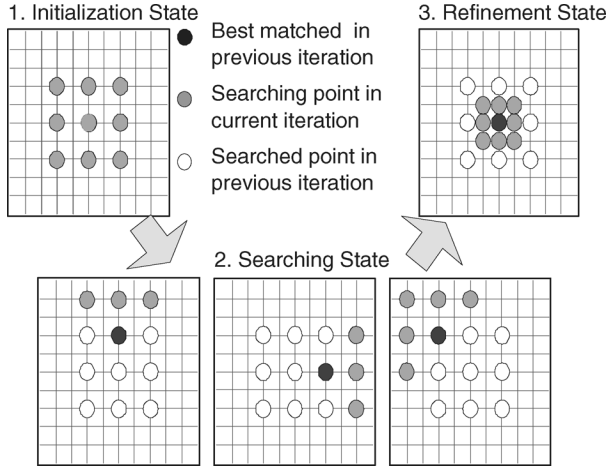


Fig. 2. Searching flow of 4SS.

irregular searching flows. The proposed architecture supporting fast algorithms should have similar DR efficiency compared with architectures supporting the FS algorithm.

III. PROPOSED HARDWARE-ORIENTED FAST ALGORITHM

Here, a hardware-oriented fast algorithm is proposed for H.264/AVC IME. Both the inter-candidate and intra-candidate DR schemes are considered. In addition, the content adaptivity is applied to achieve good tradeoff between compression performance and computational complexity.

A. DR and Content Adaptation

The DR concept is very important for a hardware-oriented fast algorithm. Two candidate-level DR schemes are considered. First, in order to achieve efficient inter-candidate DR, a rectangular search pattern, just like FS, is a better choice. Therefore, the 4SS is chosen as the base of our fast algorithm. Fig. 2 shows the searching flow of 4SS. In the initialization state, 3×3 candidates with steps of two pixels are searched. In the searching state, the search pattern moves according to the best match of the previous iteration. Finally, if the best matched candidate is the central point, the refinement is performed around the neighboring eight candidates.

Besides the inter-candidate DR, the intra-candidate DR is also utilized. In the previous works, the 4SS searching flow may repeat 41 times for 41 variable blocks. In our algorithm, the 4SS searching flow is performed only for 16×16 block. All costs of variable blocks are generated online within the 16×16 block. The moving flow follows the minimum cost of the 16×16 block. The intra-candidate DR applied in 4SS is called parallel-VBS 4SS.

However, when multiple objects move along different directions, the parallel-VBS strategy cannot accurately trace the motion vectors (MVs) of smaller blocks and may lead to some quality drop. Fig. 3 shows an example. In this scene, the moon is still, and the cloud is moving. It is hard to trace the best match of 16×8 partitions because the searching flow will be trapped in a local minimum of 16×16 block. In order to provide a robust coding efficiency for VBS-ME, more candidates should be searched in this situation.

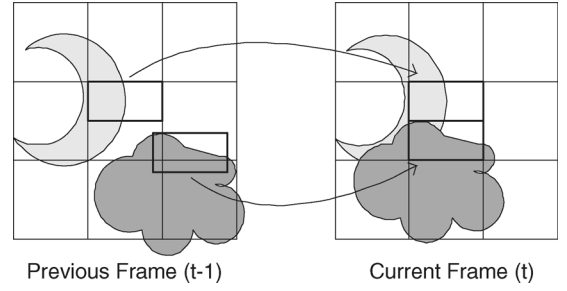


Fig. 3. Example of the complex motion scene. The moon is still, and the cloud is moving.

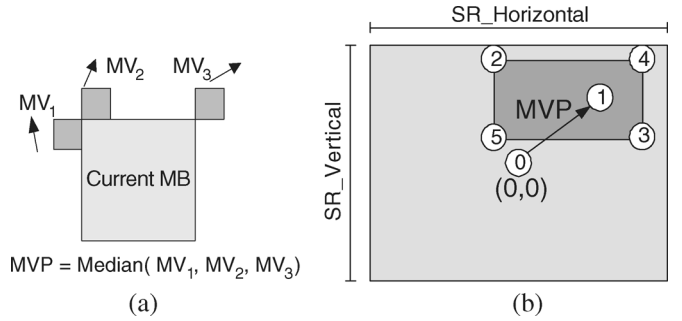


Fig. 4. Content adaptation by use of the neighboring motion activity. (a) MVP and the corresponding neighboring MVs. (b) Initial points expanded according to neighboring motion activity for tracing accurate motions of VBS.

The neighboring motion activities can be exploited to achieve a good tradeoff between the compression performance and the number of searched candidates. The MV predictor (MVP) shown in Fig. 4(a) is generally used as the initial search center to utilize the spatial correlation between neighboring MBs. The MVP is the median of left, up, and up-right blocks' MVs. If these neighboring MVs are quite different, there should be several objects moving toward different directions. In this situation, more initial points are generated according to these MVs. In this way, the different objects can be accurately traced. In general, when the motion activity is more complex, we should search more candidates to avoid the quality drop.

B. Procedure of Content-Adaptive Parallel-VBS 4SS

Based on these concepts, the content-adaptive parallel-VBS 4SS algorithm is proposed as shown in Fig. 5. At first, the MVs of the neighboring blocks, MV_1 , MV_2 , and MV_3 in Fig. 4(a), are exploited to generate the multiple initial search centers. As Fig. 4(b) shows, except for MVP, there will be four additional initial search centers, and these search centers form a window. Four boundaries of this window are calculated as follows:

$$\begin{aligned} \text{Bound}_{\text{up}} &= \max(MV_{y1}, MV_{y2}, MV_{y3}) + \text{Step} \\ \text{Bound}_{\text{down}} &= \min(MV_{y1}, MV_{y2}, MV_{y3}) - \text{Step} \\ \text{Bound}_{\text{left}} &= \min(MV_{x1}, MV_{x2}, MV_{x3}) - \text{Step} \\ \text{Bound}_{\text{right}} &= \max(MV_{x1}, MV_{x2}, MV_{x3}) + \text{Step}. \end{aligned}$$

Next, the number of the initial search centers will be adjusted according to the motion activity. If the horizontal components of MVs are similar, that means only vertical motion is involved,

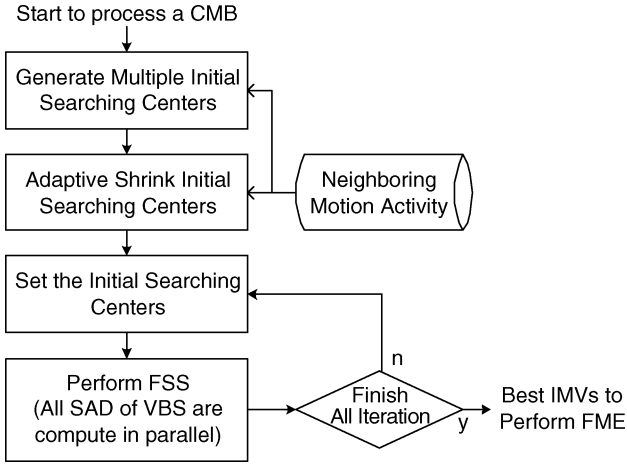


Fig. 5. Procedure of the proposed content-adaptive parallel-VBS 4SS algorithm.

and vice versa. Therefore, the expended initial search centers can be shrunk according to the following conditions:

$$\begin{aligned}
 &\text{If } (|MV_{x1} - MV_{x2}| \leq TH \ \& \ |MV_{x2} - MV_{x3}| \leq TH \ \& \\
 &\quad |MV_{x1} - MV_{x3}| \leq TH) \\
 &\quad \text{Bound}_{\text{left}} = \text{Bound}_{\text{right}} = \text{median}(MV_{x1}, MV_{x2}, MV_{x3}) \\
 &\text{If } (|MV_{y1} - MV_{y2}| \leq TH \ \& \ |MV_{y2} - MV_{y3}| \leq TH \ \& \\
 &\quad |MV_{y1} - MV_{y3}| \leq TH) \\
 &\quad \text{Bound}_{\text{up}} = \text{Bound}_{\text{down}} = \text{median}(MV_{y1}, MV_{y2}, MV_{y3}).
 \end{aligned}$$

Because background with zero motion usually occurs, we always need to add the origin as another initial search center. In the case that both conditions are satisfied, only the MVP and origin are set as the initial search centers. Finally, the 4SS performs several times according to the number of selected initial search centers. All costs of VBS are calculated in parallel with intra-candidate DR. The 41 best integer MVs are generated after all iterations are finished. Note that the two parameters of Step and TH are decided empirically and are varied with the different video specifications.

In summary, the content-adaptive parallel-VBS 4SS algorithm is proposed for the low-power hardwired IME engine. 4SS having the rectangular search pattern is suitable for hardware to reuse reference pixels between adjacent candidates. The memory accessing power can be greatly reduced with this inter-candidate DR. The parallel-VBS 4SS processes variable blocks simultaneously with 16×16 -block 4SS to reuse 4×4 costs for larger blocks. Both the memory accessing power and computational power can be saved with this intra-candidate DR. In addition, fast algorithms usually have considerable quality drop when the searching process is trapped in the local minimum. The quality drop can be compensated with more initial candidates, which greatly increases the computation complexity. The content adaptivity that adjusts the number of initial candidates according to the neighboring motion activity is applied to achieve a good tradeoff between compression performance and computation complexity. The simulation results will be shown in Section V.

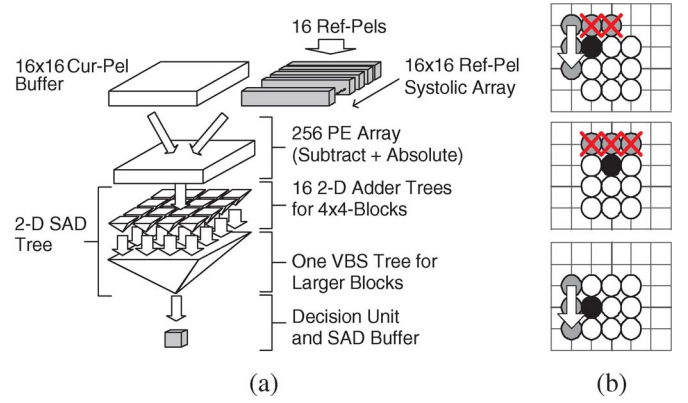


Fig. 6. (a) 2-D SAD tree architecture [11] supporting both FS and 4SS. (b) DR problem for 4SS.

IV. ARCHITECTURE DESIGN

Here, a parallel architecture is designed to support the proposed content-adaptive parallel-VBS 4SS algorithm. The 2-D adder tree architecture is used to support the intra-candidate DR. The ladder-shaped SW data arrangement and the advanced searching flow are proposed to achieve efficient inter-candidate DR.

A. Parallel Hardware With Inter-Candidate Data Reuse

Most of the previous IME architectures supporting fast algorithms have poor inter-candidate DR. Here are two examples that support the 4SS algorithm. For simplification, the interval of the square pattern in 4SS is defined as one pixel in this section.

Fig. 6(a) shows the “2-D SAD Tree” architecture [11] that supports both FS and 4SS. The CMB is stored in “ 16×16 Cur-Pel Buffer.” A row of 16 reference pixels is input and shifted downward in “ 16×16 Ref-Pel Systolic Array” in each cycle. In this way, the inter-candidate DR can be achieved between vertically adjacent candidates. Residues are generated in “256-PE Array” and then summed up by “2-D SAD Tree.” For the FS algorithm, after the latency of 15 cycles, this architecture can process one candidate for each cycle, and each candidate requires 16 reference pixels read from memories in average. For the 4SS algorithm, the reference pixels can be reused only for vertically adjacent candidates, which is shown in Fig. 6(b). For the horizontally adjacent candidates marked by “X,” each of them requires 256 reference pixels and 16 cycles. Therefore, 1856 ($16 \times 4 + 256 \times 7$) pixels are required for the 11 gray candidates in Fig. 6(b). On average, 169 reference pixels are required for each candidate. In addition, the hardware utilization and throughput largely decrease for the latency cycles.

Fig. 7(a) shows the “Parallel 1-D Tree” architecture that is also developed for FS [25] and 4SS [15] algorithms. Eighteen reference pixels and 16 CMB pixels are broadcast to the three “1-D 16 PE Arrays.” Sixteen cycles are required to process three horizontally adjacent candidates in parallel. For the FS algorithm, the reference pixels can be reused by the three horizontal candidates, and 96 ($18 \times 16/3$) pixels are required for each candidate. For the 4SS algorithm, there is a DR problem for vertically adjacent candidates, as shown in Fig. 7(b). 1856 ($96 \times 6 +$

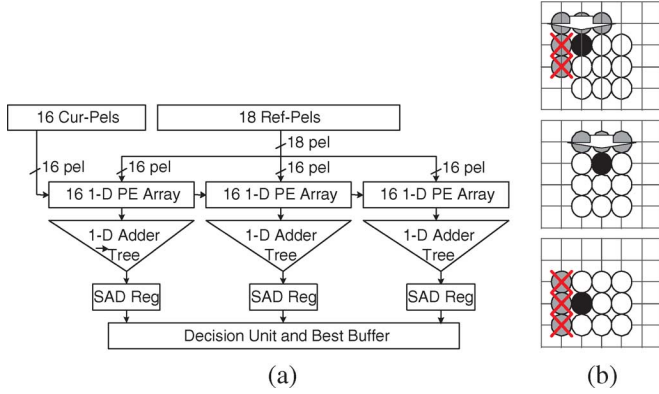


Fig. 7. (a) Parallel 1-D tree architecture supporting both FS [25] and 4SS [15]. (b) DR problem for 4SS.

256 × 5) pixels are required for 11 gray candidates. In average, 169 reference pixels are required for each candidate.

B. Proposed Techniques for Inter-Candidate DR

We start from the “2-D Adder Tree” rather than the “Parallel 1-D Tree” as the basic architecture. Three reasons are stated as follows. First, because of the systolic array structure with larger degrees of parallelism, the “2-D Adder Tree” architecture potentially has better DR capability. Second, the “1-D Tree” architecture usually co-works with the partial distortion elimination (PDE) algorithm [26] that can terminate the unnecessary computation by comparing the partial and minimum SAD costs. However, to support the intra-candidate DR, the costs of 4×4 blocks are reused for the larger blocks. The PDE cannot be efficiently applied in this situation. Third, the “2-D Adder Tree” architecture can support intra-candidate DR without partial SAD registers [10]. This hardware overhead is largely required by the “Parallel 1-D Tree.”

As for the inter-candidate DR problem to support fast algorithms, it mainly comes from the access restriction in SW SRAMs. Fig. 8(a) shows the physical location of the reference pixels in SW. In tradition, the horizontally adjacent pixels are interleavingly arranged in different SW SRAMs. As shown in Fig. 8(b), the first column of reference pixels is placed in the memory “M1.” The second column is placed in the memory “M2,” and so on. If there are eight memories, the ninth column is placed in the following entries after the first column in the memory “M1.” In this way, a row of reference pixels, as “A5–H5” in Fig. 8(b), can be read in parallel. However, a column of reference pixels, as “C1–C8” in Fig. 8(b), cannot be accessed in parallel. It is defined as the 1-D random access.

The ladder-shaped SW data arrangement is proposed to support the 2-D random access. As shown in Fig. 8(c), the second, third, fourth, and the following rows are rotated rightward by one, two, three, and the remaining pixels. In this way, the reference pixels of “A5–H5” and “C1–C8” are both arranged in different memories. Both the horizontally and vertically adjacent reference pixels can be accessed in parallel, which is the 2-D random access. For the FS algorithm, because the searching flow is regular, the 1-D random access can efficiently support inter-candidate DR. However, for fast algorithms, the search

pattern can move with various directions, and the 1-D access is not enough. With the ladder-shaped SW data arrangement, both the horizontally and vertically adjacent reference pixels can be read in parallel.

To support inter-candidate DR with 2-D random access, the “16 × 16 Ref-pel Systolic Array” in Fig. 6(a) is designed with four configurations: up-shift, down-shift, left-shift, and right-shift by one pixel. In addition, there are 16 memories, and each memory has 8-b output bit-width. The reference pixels are placed in these memories with ladder-shaped SW data arrangement. Fig. 9 shows an example of 4SS searching flow. The dotted line represents the basic flow. In Step-2, the systolic array is configured as an up-shift configuration. The corresponding rows of reference pixels are read, and totally 18 (16 + 1 + 1) cycles are required. In Step-3, the systolic array is firstly set as an up-shift configuration, and the reference pixels are read row by row, just like for Step-2. After 18 cycles, the systolic array is changed to a left-shift configuration. The corresponding two columns of reference pixels are read in the next two cycles, and two horizontally adjacent candidates can be immediately processed. Totally 20 (16 + 1 + 1 + 1 + 1) cycles are required for Step-3. In Step-4, the inter-candidate DR can be achieved with a right-shift configuration. 18 (16 + 1 + 1) cycles are required.

Although the inter-candidate DR can be achieved in both the horizontal and vertical directions, the DR rate and hardware utilization are still limited by the long latency cycles in the start of each step. Therefore, the advanced searching flow is proposed as the solid line in Fig. 9. The concept is stated as follows. Because the inter-candidate DR can be supported for any pairs of adjacent candidates, we just try to string up all required candidates. Different from the previous fast algorithms that will skip the searched candidates as many as possible, we utilize this redundant computation to tightly connect the searching flow of each step. Though the bubble cycles will occur, the long latency cycles can be eliminated. After Step-1 in Fig. 9, the reusable data are stored in “16 × 16 Ref-pel Systolic Array.” We use two bubble cycles to load two additional columns of reference pixels, and Step-2 can be immediately processed in the third cycle. The systolic array is first set as right-shift configuration for three cycles and then changed to up-shift configuration for two cycles. Similarly, after Step-2, one bubble cycle is used to load one row of reference pixels, and Step-3 can be immediately processed afterward. The systolic array is set as down-shift for one cycle, right-shift for one cycle, up-shift for two cycles, and left-shift for two cycles. In this example, 38 (24 + 5 + 6 + 3) cycles in total are required for the advanced flow, while 80 (24 + 18 + 20 + 18) cycles are required for the basic flow.

C. Architecture Design With ROM-Based Control Core

Fig. 10 shows the block diagram of the proposed architecture. The data path is very similar to Fig. 6(a) except that the systolic array has four configurations. As for the control part, in order to support the 2-D random access and the advanced searching flow, a ROM-based 4SS control core is designed. The “Moving Direction ROM” can output the moving direction according to three parameters—the end-point (EP) and minimum-point (MP)

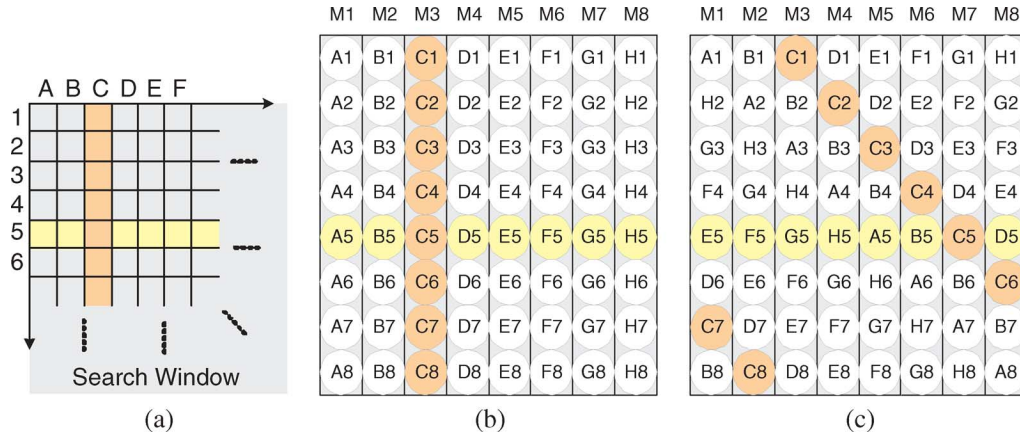


Fig. 8. (a) Physical location of SW. (b) Traditional interleaving SW data arrangement supporting 1-D random access. (c) Proposed ladder-shaped SW data arrangement supporting 2-D random access.

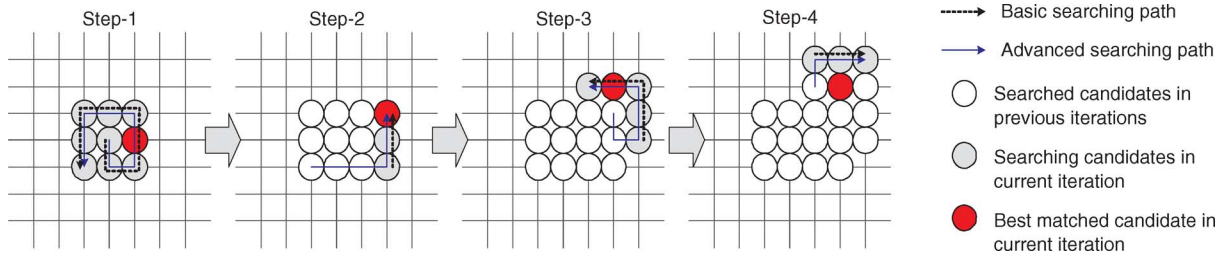


Fig. 9. Basic searching flow and advanced searching flow with 2-D random access for 4SS.

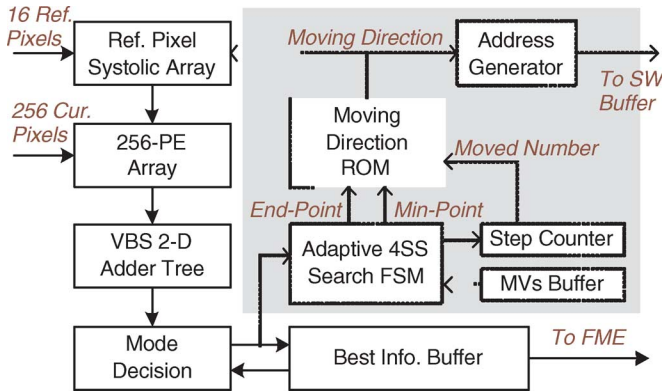


Fig. 10. Block diagram of the proposed low-power IME architecture. The 2-D random access and the advanced searching flow are operated simultaneously with ROM-based control core.

of the previous step, and the moved-number (MN) of the current step. Taking Step-2 in Fig. 9 as an example, the EP of the previous step is the bottom-left point, and the MP is the right point. When Step-2 begins to be processed, the “Step Counter” is reset to zero and then counts up by one every cycle. With the increase of the MN, the ROM will sequentially output signals as right, right, right, up, and up. Then, the address generator and the systolic array operate according to the moving directions. The EP can have four cases of left-top, left-bottom, right-top, and right-bottom. The MP can be one of the eight candidates in the 3×3 square search pattern except for the center. The maximum number of MN is eight in the case, for example, when EP is in the left-bottom point and the MP is in the right-top point.

The ROM size is $4 \times 8 \times 8$, which are the maximum numbers of EP, MP, and MN, respectively.

V. SIMULATION AND IMPLEMENTATION RESULTS

A. Performance of the Proposed Hardware Oriented Fast Algorithm

The proposed algorithm is implemented by modifying the JM8.2 encoder. Table I summarizes the reduction in computational complexity. Although VBS-ME with the FS algorithm can achieve the highest compression performance, the required computational complexity is too high even with the intra-candidate DR strategy. Fast algorithms are essential for resource-constrained mobile devices, and 4SS is chosen for its potential of inter-candidate DR in hardware implementation. The “sequential-VBS 4SS,” which sequentially processes the 41 variable blocks, limits the computational saving. The “single-iteration parallel-VBS 4SS” performs 4SS on the 16×16 block and generates the costs of smaller blocks in parallel. Because of intra-candidate DR, the computational complexity is reduced to about 1/7, but a considerable quality drop is induced especially for the sequences with a complex motion activity. The proposed “multi-iteration parallel-VBS 4SS” extracting more initial search centers can both maintain the VBS performance and achieve parallel processing for variable blocks. After the technique of content adaptivity is included, a good tradeoff between computation reduction and compression performance can be achieved. Note that the parameters of Step and TH are decided empirically according to the software simulations and are both set to two pixels for CIF specifications.

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISON BETWEEN FS AND FAST ALGORITHMS

Unit: 16×16 -sized Candidates per CMB

	Parallel-VBS Full Search	Sequential-VBS 4SS	Parallel-VBS 4SS		
			Single-iteration	Multi-iteration	Content-Adaptive
Stefan	2048	124.05	21.18	114.49	61.10
Foreman	2048	160.76	23.68	112.02	67.50
Mobile	2048	119.77	17.34	103.90	51.90
Silent	2048	127.68	18.24	103.02	33.82
Coastguard	2048	122.65	17.95	102.08	43.63

Coding parameters: H.264 Baseline Profile, JM8.2, IPP..., 1 ref. frame, H.SR = ± 32 , V.SR = ± 16 , QP = 15, 18,..., 42

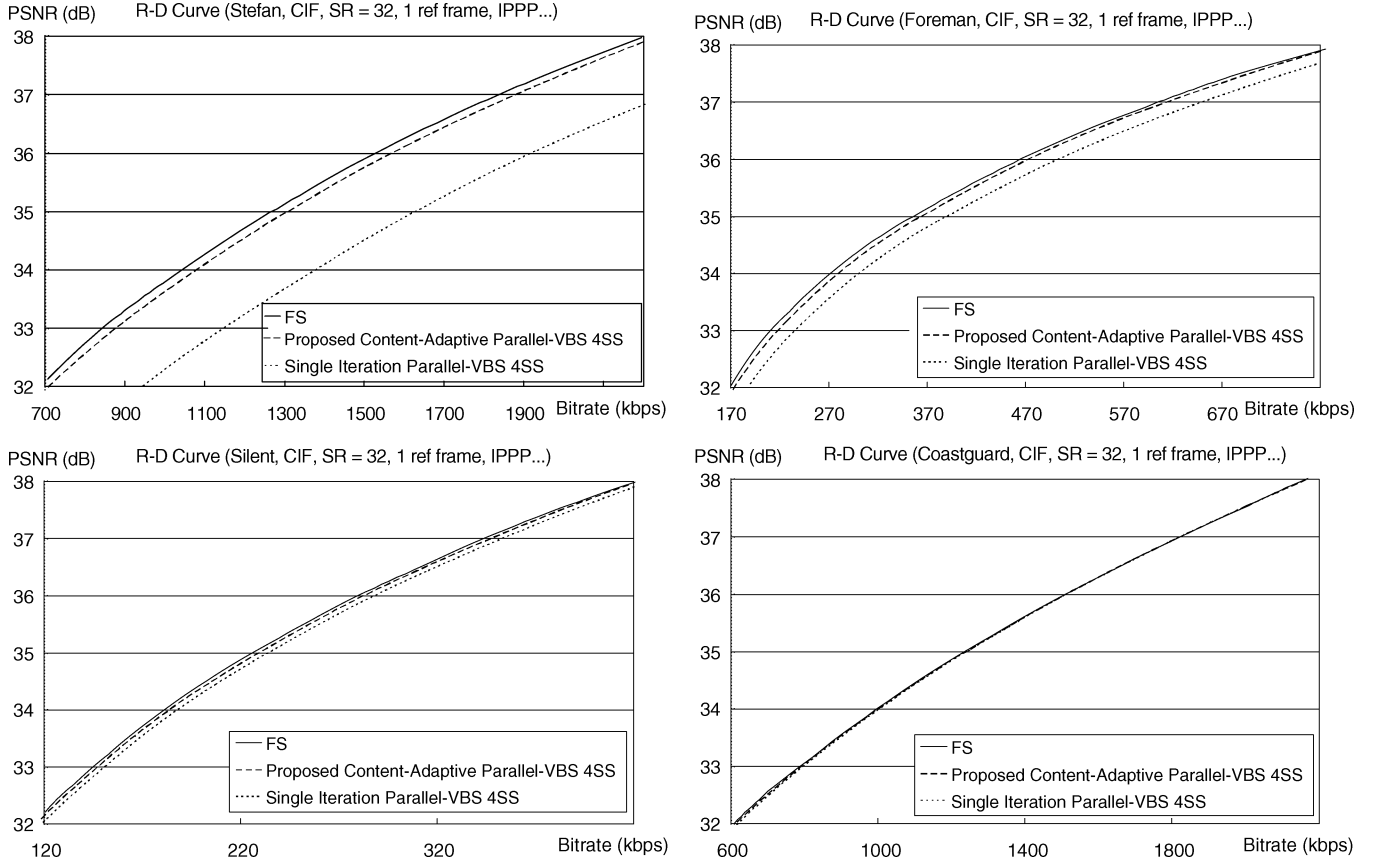


Fig. 11 Comparisons of the rate-distortion efficiency between FS and fast algorithms.

Fig. 11 shows the rate distortion efficiencies of the FS, proposed “content adaptive parallel-VBS 4SS,” and “single-iteration parallel-VBS 4SS” algorithms. The proposed algorithm is robust even for the video with a high motion activity (stefan).

B. Performance of the Proposed Architecture for Inter-Candidate DR

One redundancy access (RA) factor can be used to evaluate the performance of DR and is defined as follows:

$$RA = \frac{\text{Number of ref-pels read from SW SRAM}}{\text{minimum requirement}}$$

The minimum requirement, or minimum number of required reference pixels, is the pixel number of the union of all searched candidates. For one candidate, the minimum requirement is 256

TABLE II
COMPARISON OF THE PERFORMANCE OF THE PROPOSED TECHNIQUES

Architecture	Parallel 1-D tree (Fig. 7)		2-D tree (Fig. 6)		
	1D	2D	1D	2D	2D
Random Access	1D	2D	1D	2D	2D
Advanced Flow	n/a	n/a	n/a	No	Yes
RA _{cand.-level}	8.77	5.09	6.86	3.23	1.54

pixels. For two horizontally or vertically adjacent candidates, the minimum requirement is 272 ($256 + 16$) pixels. If the RA factor is two, this means the number of read pixels is twice the minimum requirement. Note that the searching flow and the search pattern shown in Fig. 9 are used as the model for the following comparison. The minimum required reference pixels in this case are 395 pixels for the 20 searched candidates. The comparison is shown in Table II. In general, the “2-D Tree” architecture has better DR efficiency than the “Parallel 1-D Tree” archi-

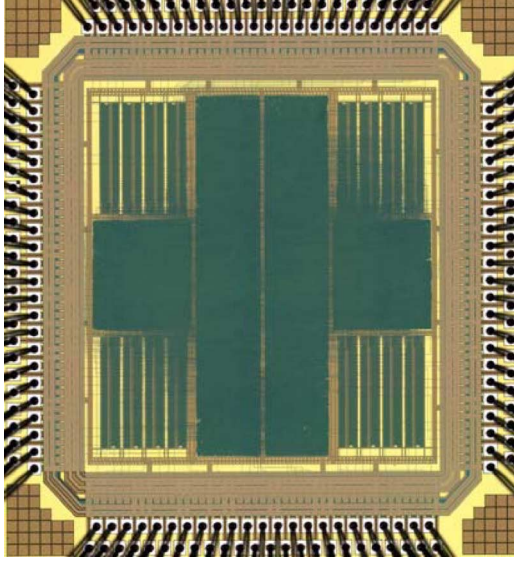


Fig. 12. Chip photograph of the proposed H.264/AVC IME engine.

TABLE III
SPECIFICATION OF THE PROPOSED H.264/AVC H.264/AVC IME ENGINE

Technology	TSMC 0.18 μm 1P6M CMOS
Pad/Core Voltage	3.3/1.8 V
Core Area	1.90 \times 1.90 mm^2
Logic Gates	131.2 K (2-input NAND gate)
SRAM	64 Kb
Max Operating Frequency	66 MHz
Fast Algorithm	Content-adaptive Parallel-VBS FSS
Block Sizes	16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4
Max. # of Ref. Frames	2
Max. SR	H[-32,+31] V[-16,+15]

ture does. The 2-D random access can support the inter-candidate DR for both horizontal and vertical directions, while the advanced searching flow can further reduce the latency cycles. After the 2-D random access and the advanced searching flow are applied, 77.6% (1–1.54/6.86) bandwidth and power of SW SRAMs are saved for the “2-D Tree” architecture.

C. Implementation Results

The proposed IME architecture is implemented on a 3.42-mm² die with TSMC 0.18- μ 1P6M technology. Fig. 12 shows the chip photograph, and the detailed chip features are listed in Table III. The total logic gate count is 131.2 K with 64-kb SRAMs. The maximum operating frequency is 40 MHz. This design can support real-time encoding CIF 30-fps videos with three modes, and the SRs are ± 32 pixel horizontally and ± 16 pixel vertically. In high-quality mode, the coding parameter is the proposed content-adaptive parallel-VBS 4SS algorithm with two reference frames. In this mode, the SW SRAMs are configured as level-C MB-level DR scheme [29]. In low-power mode, the coding parameter is the content-adaptive parallel-VBS 4SS with one reference frame. Since only one SW is required in this mode, the SW SRAMs are configured as the level-D MB-level DR scheme [29] to achieve the minimum system bandwidth for the lower power consumption of the whole system. In ultralow-power mode, the single-iteration parallel-VBS 4SS algorithm is used. This

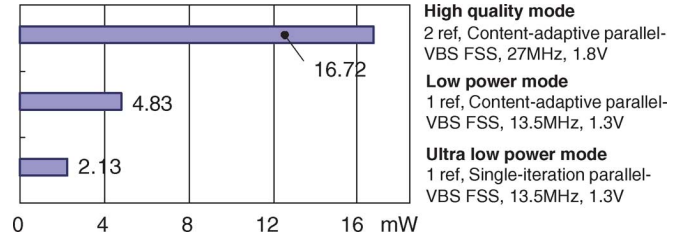


Fig. 13. Power consumption results of the proposed architecture.

means that only the MVP is used as the initial search center. The operation frequency is 27 MHz with 1.8-V supply voltage for the high-quality mode and 13.5 MHz with 1.3 V for the remaining two modes.

Fig. 13 shows the measured power consumption of this chip. Because the average computational complexity is generally lower than the worst case, the operating frequency is decided according to the worst case. The gated clock technique is implemented to turn the inoperative circuits off when IME sleeps. In addition, in the low-power and ultralow-power modes, the computational complexity is reduced, and so is the operating frequency. When the operating frequency is 13.5 MHz, the voltage scaling-down technique can be used to further reduce the power consumption. For real-time encoding CIF 30-fps videos, in high-quality mode, the power consumption is 16.72 mW with a similar compression performance compared with the FS algorithm. In the ultralow-power mode, the power consumption can be as small as 2.13 mW.

The comparison with the previous methods are listed in Table IV. Because they are all designed for the previous standards, where VBS and MRF are not supported, the parameter of our design is set as the single-iteration 4SS with one reference frame. Since different processes and supply voltages are used, we normalize the power data according to the supply voltage and the dimension for the comparison. Chao's and J.M's designs use the 1-D tree architecture without any inter-candidate DR. Huang's design uses the global elimination fast algorithm with global search pattern and has related high computation complexity. Therefore, these three designs require higher power consumption. As for Lin's design, it uses the parallel 1-D tree architecture supporting the inter-candidate DR among horizontally adjacent candidates. The proposed architecture with the 2-D tree architecture supports the inter-candidate DR for both horizontally and vertically adjacent candidates. It can reuse data in the most efficient way and therefore has the lowest power consumption.

VI. CONCLUSION

In this paper, a parallel architecture with efficient DR techniques and a hardware-oriented algorithm is proposed for low-power H.264/AVC IME. According to our analysis, the power consumption of IME module mainly comes from two parts: the data access power and the computational power. A content-adaptive parallel-VBS 4SS algorithm is first designed with the inter-/intra-candidate DR capability for hardware implementation, and 97% computational complexity is saved. Then, based on the systolic array and 2-D adder tree architecture, a ladder-

TABLE IV
COMPARISON OF POWER CONSUMPTION AMONG OUR ARCHITECTURE AND THE PREVIOUS METHODS

	[13,27] Chao's	[28] Huang's	[14] J.M.'s	[15] Lin's	This Work
Process (μm)	0.35	0.35	0.18	0.18	0.18
Voltage (V)	3.3	3.3	1.0	1.8	1.3
Core Size	5.2	7.5	13.7	2.2	3.6
Transistors Count	301K	358K	1000K	546K	708K
Max Frequency (MHz)	50	27.8	13.5	48.67	66
Video Spec.	CIF 30-fps	CIF 30-fps	CIF 30-fps	CIF 30-fps	CIF 30-fps
Frequency (MHz)	50	25	13.5	48.67	13.5
Algorithm	Diamond Search	Global Elimination	Gradient Decent	4SS	Single-iteration Parallel-VBS 4SS w/ 1-ref.
Block Size	16 \times 16 and 8 \times 8	16 \times 16	16 \times 16 and 8 \times 8	16 \times 16	16 \times 16 to 4 \times 4
Power (mW)	223.6	189	6.56	8.46	2.13
Normalized Power (1.8V, 0.18 μm)(*)	17.60	14.87	21.25	8.46	4.08

$$(*) \text{ NormalizedPower} = \text{Power} \times \frac{0.18^2}{\text{Process}^2} \times \frac{1.8^2}{\text{Voltage}^2}$$

shaped SW data arrangement and advanced searching flow are applied to support inter-candidate DR and to reduce the latency cycles. Memory bandwidth is reduced by 77.6%. According to the implementation results, the power consumption is 2.13 mW for real-time encoding CIF 30-fps videos at 13.5-MHz operating frequency.

REFERENCES

- [1] Joint Video Team, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, May 2003.
- [2] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.
- [3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [4] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, pp. 7–28, 2004.
- [5] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," *Signal Process.: Image Commun.*, vol. 19, pp. 793–849, Oct. 2004.
- [6] T.-C. Chen, S.-Y. Chien, Y.-W. Huang, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, and L.-G. Chen, "Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 6, pp. 673–688, Jun. 2006.
- [7] H.-C. Chang, L.-G. Chen, M.-Y. Hsu, and Y.-C. Chang, "Performance analysis and architecture evaluation of MPEG-4 video codec system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2000, vol. 2, pp. 449–452.
- [8] J.-H. Lee and N.-S. Lee, "Variable block size motion estimation algorithm and its hardware architecture for H.264," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2004, vol. 3, pp. 740–743.
- [9] Y.-W. Huang, T.-C. Wang, B.-Y. Hsieh, and L.-G. Chen, "Hardware architecture design for variable block size motion estimation in MPEG-4 AVC/JVT/ITU-T H.264," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2003, vol. 2, pp. II796–II799.
- [10] S. Y. Yap and J. V. McCanny, "A VLSI architecture for variable block size video motion estimation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 7, pp. 384–389, Jul. 2004.
- [11] C.-Y. Chen, S.-Y. Chien, Y.-W. Huang, T.-C. Chen, T.-C. Wang, and L.-G. Chen, "Analysis and architecture design of variable block size motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 3, pp. 578–593, Mar. 2006.
- [12] J. Miyakoshi, Y. Murachi, K. Hamano, T. Matsuno, M. Miyama, and M. Yoshimoto, "A low-power systolic array architecture for block-matching motion estimation," *IEICE Trans. Electron.*, pp. 559–569, 2005.
- [13] W.-M. Chao, C.-W. Hsu, Y.-C. Chang, and L.-G. Chen, "A novel hybrid motion estimator supporting diamond search and fast full search," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2002, vol. 2, pp. II-492–II-495.
- [14] J. Miyakoshi, Y. Kuroda, M. Miyama, K. Imamura, H. Hashimoto, and M. Yoshimoto, "A sub-mW MPEG-4 motion estimation processor core for mobile video application," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2003, pp. 181–184.
- [15] S.-S. Lin, "Low-Power Motion Estimation Processors for Mobile Video Application," M.S. thesis, Graduate Inst. of Electron. Eng., Nat. Taiwan Univ., Taipei, Taiwan, R.O.C., 2004.
- [16] J. C. Tuan, T. S. Chang, and C. W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61–72, Jan. 2002.
- [17] T.-C. Chen, Y.-W. Huang, C.-Y. Tsai, C.-T. Huang, and L.-G. Chen, "Single reference frame multiple current macroblocks scheme for multi-frame motion estimation in H.264/AVC," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, vol. 2, pp. 1790–1793.
- [18] H. F. Ates and Y. Altunbasak, "SAD reuse in hierarchical motion estimation for the H.264 encoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2005, pp. II-905–II-908.
- [19] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438–442, Aug. 1994.
- [20] 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, no. 3, pp. 313–317, Jun. 1996.
- [21] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–377, Aug. 1998.
- [22] M.-H. Chan, Y.-B. Yu, and A.-G. Constantinides, "Variable size block matching motion compensation with applications to video coding," in *Proc. Inst. Elect. Eng. Commun., Speech Vis.*, Aug. 1990, vol. 137, pp. 205–212.
- [23] I. Rhee, G. R. Martin, S. Muthukrishnan, and R. A. Packwood, "Quadtree-structured variable-size block-matching motion estimation with minimal error," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 1, pp. 42–50, Feb. 2000.
- [24] Z. Zhou, M.-T. Sun, and Y.-F. Hsu, "Fast variable block-size motion estimation algorithm based on merge and split procedures for H.264/MPEG-4 AVC," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2004, vol. 3, pp. 725–728.
- [25] P.-C. Tseng, S.-S. Lin, and L.-G. Chen, "Low-power parallel tree architecture for full-search block-matching motion estimation," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2004, pp. 239–244.
- [26] Telenor R&D, ITU-T Recommendation H.263 Software Implementation Digital Video Coding Group, 1995.
- [27] W.-M. Chao, "Platform-based design and chip implementation of MERG-4 video coding," M.S. thesis, Graduate Inst. Electron. Eng., Nat. Taiwan Univ., Taipei, Taiwan, R.O.C., 2002.
- [28] Y.-W. Huang, S.-Y. Chien, B.-Y. Hsieh, and L.-G. Chen, "Global elimination algorithm and architecture design for fast block matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 898–907, Jun. 2004.
- [29] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61–72, Jan. 2002.



Tung-Chien Chen was born in Taipei, Taiwan, R.O.C., in 1979. He received the B.S. degree in electrical engineering and the M.S. degree in electronic engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 2002 and 2004, respectively, where he is working toward the Ph.D. degree in electronics engineering.

His major research interests include motion estimation, algorithm and architecture design of MPEG-4 and H.264/AVC video coding, and low-power video coding architectures.



Yu-Han Chen was born in Taipei, Taiwan, R.O.C., in 1981. He received the B.S. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., in 2003. He currently is working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering, National Taiwan University.

His research interests include image/video signal processing, motion estimation, algorithm and architecture design of H.264 video coder, and low-power and power-aware video coding system.



Sung-Fang Tsai was born in Hsinchu, Taiwan, R.O.C., in 1983. He received the B.S. degree in electrical engineering in electronic engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 2005. He is currently working toward the M.S. degree at the Graduate Institute of Electronics Engineering, National Taiwan University.

His major research interests include motion estimation and algorithm and architecture design of H.264/AVC video coding standard.



Shao-Yi Chien was born in Taipei, Taiwan, R.O.C., in 1977. He received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 1999 and 2003, respectively.

During 2003 to 2004, he was a Member of Research Staff with the Quanta Research Institute, Tao Yuan Shien, Taiwan, R.O.C. In 2004, he joined the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, as an Assistant Professor. His

research interests include video segmentation algorithm, intelligent video coding technology, image processing, computer graphics, and associated VLSI architectures.



Liang-Gee Chen (S'84–M'86–SM'94–F'01) was born in Yun-Lin, Taiwan, R.O.C., in 1956. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 1979, 1981, and 1986, respectively.

He was an Instructor (1981–1986) and an Associate Professor (1986–1988) with the Department of Electrical Engineering, National Cheng Kung University. During his service in the military during 1987 and 1988, he was an Associate Professor with the Institute of Resource Management, Defense Management College. In 1988, he joined the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. From 1993 to 1994, he was a Visiting Consultant with the DSP Research Department, AT&T Bell Laboratories, Murray Hill, NJ. In 1997, he was a Visiting Scholar with the Department of Electrical Engineering, University of Washington, Seattle. Currently, he is a Professor with National Taiwan University. Since 2004, he has also been the Executive Vice President and the General Director of Electronics Research and Service Organization (ERSO) in the Industrial Technology Research Institute (ITRI). His current research interests are DSP architecture design, video processor design, and video coding system.

Dr. Chen is a member of Phi Tan Phi. He was the General Chairman of the 7th VLSI Design CAD Symposium and the 1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from June 1996 until now and as an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE-SCALE INTEGRATED (VLSI) SYSTEMS from January 1999 until now. He was an Associate Editor for the *Journal of Circuits, Systems, and Signal Processing* from 1999 until now. He served as the Guest Editor of the *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology* in November 2001. He is also an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: EXPRESS BRIEFS. In 2002, he became an Associate Editor of the PROCEEDINGS OF THE IEEE. He was the recipient of the Best Paper Award from ROC Computer Society in 1990 and 1994. From 1991 to 1999, he was the recipient of the Long-Term (Acer) Paper Awards annually. In 1992, he was the recipient of the Best Paper Award of the 1992 Asia-Pacific Conference on Circuits and Systems in VLSI design track, the Annual Paper Award of Chinese Engineer Society in 1993, and the Outstanding Research Award from the National Science Council of Taiwan and the Dragon Excellence Award for Acer both in 1996. He was elected an IEEE Circuits and Systems Distinguished Lecturer from 2001–2002.