

A Content-Based Methodology for Power-Aware Motion Estimation Architecture

Hsien-Wen Cheng and Lan-Rong Dung, *Member, IEEE*

Abstract—This paper presents a novel power-aware motion estimation algorithm, called adaptive content-based subsample algorithm (ACSA), for battery-powered multimedia devices. While the battery status changes, the architecture adaptively performs graceful tradeoffs between power consumption and compression quality. As the available energy decreases, the algorithm raises the subsample rate for maximizing battery lifetime. Differing from the existing subsample algorithms, the content-based algorithm first extracts edge pixels from a macro-block and then subsamples the remaining low-frequency part. In this way, we can alleviate the aliasing problem and thus keep the quality degradation low as the subsample rate increases. As shown in experimental results, the architecture can dynamically operate at different power consumption modes with little quality degradation according to the remaining capacity of battery pack while the power overhead of edge extraction is under 0.8%.

Index Terms—Content-based image processing, motion estimation (ME), power-aware system, subsample algorithm, very large-scale integration (VLSI) architecture, video compression, VLSI image processing.

I. INTRODUCTION

MOTION ESTIMATION (ME) has been notably recognized as the most critical part in many video compression applications, such as MPEG standards and H.26x [1], which leads to dominant computational and, hence, power requirements. With increasing demand of portable multimedia devices, recently, a power-aware ME that can be flexible in both power consumption and compression quality is highly required [2]. Fig. 1 illustrates a typical block diagram of the portable multimedia system powered by battery. In practice, a battery has the two most important nonideal properties, which are the rate capacity effect and recovery effect [3]. Fig. 2, as an example, illustrates that the system can extend the battery lifetime by gradually stepping down the power dissipation. Normally, system designers can choose points A, B, C, and D according to the discharging profile provided by battery manufacturer. When the battery monitor unit detects the voltage drop of battery, the host processor will change the power mode accordingly. By changing the operation mode of all power-aware components simultaneously, the architecture adapts the power dissipation to battery status and, hence, raises battery performance.

Manuscript received February 24, 2004; revised September 13, 2004 and February 22, 2005. This work was supported in part by Taiwan MOE Program for Promoting Academic Excellent of Universities under Grant 91-E-FA06-4-4 and the National Science Council, Taiwan, R.O.C., under Grant NSC 93-2220-E-009-023. This paper was recommended by Associate Editor S.-F. Chang.

The authors are with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: jswcheng.ece89g@nctu.edu.tw; lennon@cn.nctu.edu.tw).

Digital Object Identifier 10.1109/TCSII.2005.850771

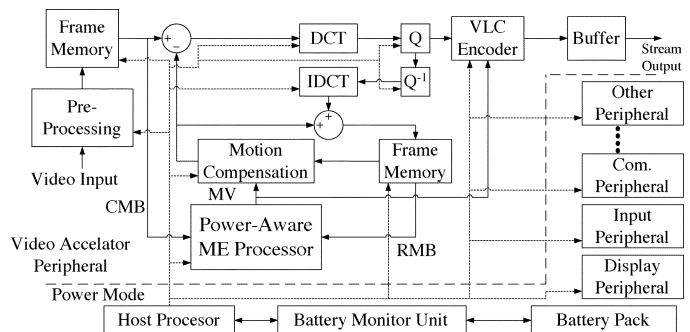


Fig. 1. System block diagram of a portable, battery-powered multimedia device.

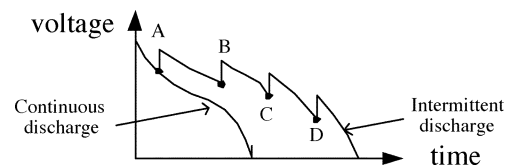


Fig. 2. Diagram representing the nonlinear discharging properties of battery.

This paper, therefore, intends to present a power-aware ME architecture driven by an adaptive content-based subsample algorithm to lengthen the battery lifetime.

Many published papers have presented efficient algorithms for very large-scale integration (VLSI) implementation of motion estimation on high performance or low-power design. Yet most of them cannot dynamically adapt the compression quality to different power consumption modes. Among these proposed algorithms, the full-search block-matching (FSBM) algorithm with the sum of absolute difference (SAD) criterion is the most popular approach for ME because of its considerably good quality. It is particularly attractive to those who require extremely high quality. However, the huge number of *comparison/difference* operations results in a high computation load and power consumption [4], [5]. To reduce the computational complexity of FSBM, researchers have proposed various fast algorithms. They either reduce search steps [6]–[8] or simplify calculations of error criterion [9], [10]. These fast-search algorithms have successfully improved the block matching speed and, thus, led to a low-power implementation. However, a low-power implementation is not necessarily a power-aware system, in which the implementation should adaptively modify its behavior to balance the performance between quality and battery life [11]. The requirement for ME algorithms to be suitable for power-aware design is a high degree of scalability in performance tradeoffs. Unfortunately, the fast algorithms mentioned above do not meet this requirement.

The subsample algorithms present in [12] and [13] are very suitable for power-aware ME architecture because of their highly scalable characteristic. As the subsample rate increasing, the unsampled processing elements in a semisystolic array will be disabled to save the switch activities, that is, the power consumption of the architecture will be decreased correspondingly. However, applying subsample algorithms for power-aware architecture may suffer from the aliasing problem in the high-frequency band. The aliasing problem degrades the compression quality rapidly as the subsample rate increases. To alleviate the problem, we extend traditional subsample algorithms to an adaptive content-based subsample algorithm (ACSA). In the ACSA, we first use edge extraction techniques to separate the high-frequency band from a macro-block and then subsample the low-frequency band only. By merging the edge pixels and subsample nonedge pixels, the algorithm generates a turn-off mask for the architecture to disable the switching activities of processing elements (PEs) in a semisystolic array. This content-based algorithm keeps the quality degradation low as the subsample rate increases. Because the number of high-frequency pixels varies with different video clips, we introduce an adaptive control mechanism to set the threshold value for edge determination and make the number of masked pixels stationary for a given power mode. The ACSA can be implemented in most existing ME architectures by turning off PEs according to the subsample mask. In this paper, we present a semisystolic architecture with gated PEs. The simulation results show that the ACSA can dynamically alter the subsample rate as the power consumption mode changes and the architecture can work at different power consumption modes with acceptable and smooth quality degradation while the power overhead of edge extraction is under 0.8%.

II. SUBSAMPLE ALGORITHMS

A. Generic Subsample Algorithm

Here, we present a generic subsample algorithm (GSA) in which the subsample rate ranges from 4:1 to 1:1. The GSA uses

$$SSAD_{8:m}(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |SM_{8:m}(i, j) (S(i+u, j+v) - R(i, j))| \quad (1)$$

as a matching criterion, called the subsample sum of absolute difference (SSAD), where $SM_{8:m}$ is the subsample mask for the subsample rate 8 to m as shown in

$$SM_{8:m}(i, j) = BM_{8:m}(i \bmod 4, j \bmod 4). \quad (2)$$

where the macro-block size is N -by- N , $R(i, j)$ is the luminance value at (i, j) of the current macro-block, and $S(i+u, j+v)$ is the luminance value at (i, j) of the reference macro-block which offsets (u, v) from the current macro-block in the searching area $2p \times 2p$. The subsample mask $SM_{8:m}$ is generated from the basic mask

$$BM_{8:m} = \begin{bmatrix} u(m-2) & u(m-5) & u(m-2) & u(m-6) \\ u(m-3) & u(m-7) & u(m-4) & u(m-8) \\ u(m-2) & u(m-5) & u(m-2) & u(m-6) \\ u(m-3) & u(m-7) & u(m-4) & u(m-8) \end{bmatrix}$$

```
//frame:t
Input current and reference frames,  $W \times H$  ;
for(y=0; y < W/N; y++){
  for(x=0; x < H/N; x++){
    Perform gradient filtering;
    Calculate the edge threshold :
     $E_{th} = m_1^t(x, y) \cdot \max\{G(i, j)\} + (1 - m_1^t(x, y)) \cdot \min\{G(i, j)\}$ 
    Determine edge pixels and edge mask;
    Generate adaptive content - based subsample mask (ACSM);
     $acsm\_cnt = total\ edges\ of\ ACSM;$ 
    //update threshold parameter for the next frame
     $m_1^{t+1}(x, y) = m_1^t(x, y) + K_p \cdot (acsm\_cnt - trg\_cnt);$ 
    if( $m_1^{t+1}(x, y) < 0$ ) { $m_1^{t+1}(x, y) = 0$ };
    if( $m_1^{t+1}(x, y) > 1$ ) { $m_1^{t+1}(x, y) = 1$ };
    //find MV
     $ACSSAD_{min}(x, y) = \infty;$ 
    for( $u = -p; u < p; u++$ ) {
      for( $v = -p; v < p; v++$ ) {
         $ACSSAD(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |ACSM(i, j) \cdot (S(i+u, j+v) - R(i, j))|;$ 
        if  $ACSSAD_{min}(x, y) > ACSSAD(u, v)$ 
          { $ACSSAD_{min}(x, y) = ACSSAD(u, v); MV(x, y) = (u, v);$  }
      }
    }
  }
}
```

Fig. 3. Procedure of the ACSA.

where

$$u(n) = 1, \quad \text{for } n \geq 0 \text{ and } 0 \text{ for } n < 0. \quad (3)$$

Due to its flexibility in energy-quality tradeoffs, the GSA is suitable for the implementation of power-aware architectures. The power consumption of the architecture is proportional to the inverse of the subsample rate. However, the algorithm suffers from the aliasing problem which will degrade the ME quality and results in a considerable degradation of quality when the high-frequency band is messed up.

B. Adaptive Content-Based Subsample Algorithm

As mentioned above, the GSA has an aliasing problem for a high subsample rate and leads to considerable quality degradation because the high-frequency band is interfered with. To alleviate the problem, we propose an ACSA that only subsamples the low-frequency band. The procedure of the ACSA is described in Fig. 3. We first use edge extraction to separate high-frequency pixels (or edge pixels) from a macro-block and then subsample the remaining pixels (or low-frequency pixels). The determination of edge pixels starts from gradient filtering. In this paper, we use three popular gradient filters to exercise the content-based algorithm; they are the high-pass gradient filter, the Sobel gradient filter, and the morphological gradient filter. After obtaining the gradients denoted as G , we use a floating threshold E_{th} to determine the edge pixels of the current macro-block. The floating threshold makes the edge extraction more robust with video content varying than the constant threshold does. The following equation describes the calculation of E_{th} for each macro-block (x, y) in the t th frame:

$$E_{th} = m^t \cdot \max\{G(i, j)\} + (1 - m^t) \cdot \min\{G(i, j)\}, \quad \text{for } 0 \leq m^t \leq 1 \text{ and } 0 \leq i, j < N. \quad (4)$$

Following the threshold setting step, the algorithm uses the threshold value to pick the edge pixels and produces the edge mask as shown as

$$\text{EdgeMask}(i, j) = \begin{cases} 1, & \text{for } G(i, j) \geq E_{\text{th}} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Finally, the adaptive content-based subsample mask (ACSM) is generated by merging the edge mask and the subsample mask, as shown in

$$\text{ACSM}(i, j) = \text{SM}_{4:1}(i, j) \vee \text{EdgeMask}(i, j), \\ 0 \leq i, j \leq N - 1. \quad (6)$$

The operator \vee means logic OR operation. According to the calculation of ACSM, the subsample rate of the ACSA R_{ACSA} is equal to N^2 to $acsm_cnt$, where $acsm_cnt$ is the number of 1's in ACSM. Once the ACSM is generated, the algorithm can then determine the motion vector (MV) with the adaptive content-based subsample sum of absolute difference (ACSSAD) criterion. The following equation shows the ACSSAD criterion:

$$\text{ACSSAD}(u, v) \\ = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\text{ACSM}(i, j) (S(i+u, j+v) - R(i, j))| \quad (7)$$

Although the content-based algorithm can perform high-power scalability and alleviate the mess in high frequency, there exists a nonstationary problem if the designer uses a constant threshold parameter m^t to statically derive the floating threshold for edge-extracting. Since different macro-blocks with the same threshold parameter will have different R_{ACSA} values, setting the threshold parameters without considering the content variation of macro-blocks will make the subsample rate nonstationary; that is, the power consumption will not be converged within a narrow range for a given power mode. For example, the subsample rate of the Weather clip can vary between 256:67 and 256:224 in the first frame while m^t is set as 0.1. To solve this problem, this paper introduces an adaptive control mechanism to adjust the threshold parameter so that the subsample rate R_{ACSA} can be stationary. The adaptive control mechanism is a run-time process that adjusts the threshold parameter fittingly according to the difference between the current subsample rate and the desired subsample rate (or target subsample rate).

Fig. 4 shows the block diagram of the adaptive control mechanism. Given the battery status, the host processor sets the power mode and the target subsample rate as well. The target subsample rate is N^2 to trg_cnt , where trg_cnt is the target number of 1's in the ACSM. Then, the controller recursively updates the threshold parameter $m^{t+1}(x, y)$ based on the current $m^t(x, y)$ and the difference of $acsm_cnt$ and trg_cnt , as shown in Fig. 3 under the line “//update threshold parameter for the next frame.”

III. POWER-AWARE ARCHITECTURE

According to the ACSA, we present a semisystolic architecture as shown in Fig. 5, based on existed architectures, such as [4]. The architecture contains an edge-extraction unit (EXU), an array of processing elements (PEs), a parallel adder tree (PAT), a shift register array (SRA), and a motion-vector selector (MVS).

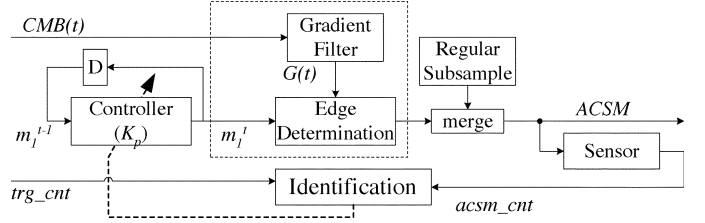


Fig. 4. Block diagram of the edge-extraction unit with adaptive control mechanism.

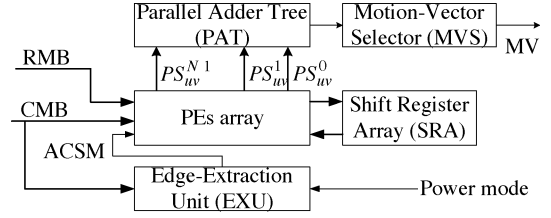


Fig. 5. Block diagram of the power-aware ME architecture driven by the ACSA.

Given the power consumption mode, the EXU, which contains the gradient filter and the ACSM generator, extracts high-frequency (or edge) pixels from the current macro-block and generates 0–1 ACSM to disable or enable PEs. The PE array is used to accumulate absolute pixel differences column by column while the parallel adder tree sums up all the results to generate the value of ACSSAD. Each PE has a datapath that consists of the absolute difference unit ($|a - b|$) and the Adder unit. Upon the ACSSAD is generated, the MVS then performs compare-and-select operation to select the best motion vector.

To start with, by performing the edge extraction, the architecture first generates ACSM to determine whether to enable/disable PEs and thus dynamically changes the switching activities of system to reduce the power consumption. The ACSM disables the PE by using a block element (BE) that is implemented by AND gates. The BEs can nullify the input signals of data path. When a PE is disabled during an MV searching iteration, the circuit in the PE remains still until the next iteration starts and, thus, the consumption of transient power can be saved.

IV. POWER MODEL

One can consider the major power consumption of a CMOS gate i as

$$P_{\text{gate}_i} = \alpha \cdot C_i \cdot f_i \cdot V_{DD}^2 = \kappa \cdot C_i \cdot r_i(\downarrow) \quad (8)$$

where C_i is the output capacitance, f_i is the operation frequency, $r_i(\downarrow)$ is the switch activity of gate i , and α and κ are constants. For an active execution unit EU_k in a VLSI system, the power consumption can be shown in

$$P_{\text{EU}_k} = \sum_{i=1}^{N_{\text{gate}, \text{EU}_k}} \kappa \cdot C_i^k \cdot r_i^k(\downarrow) \quad (9)$$

where $N_{\text{gate}, \text{EU}_k}$ is the gate count of EU_k . The following equations demonstrate the total power consumption:

$$P_{\text{total}} = \sum_{\forall \text{activeEU}_k} \kappa \sum_{i=1}^{N_{\text{gate}, k}} C_i^k \cdot r_i^k(\downarrow) \quad (10)$$

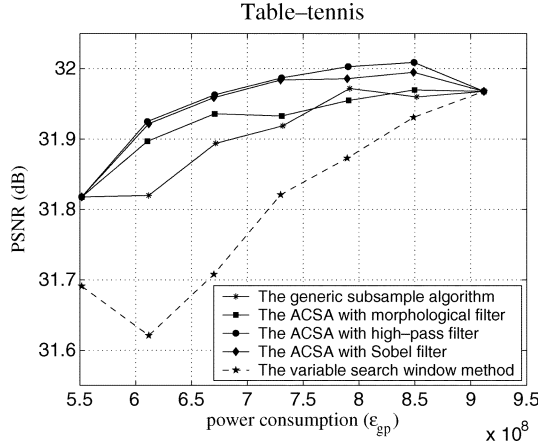


Fig. 6. Quality degradation of the “table tennis” clip.

$$\cong \kappa \sum_{\forall \text{activeEU}_k} r^k(\downarrow) \sum_{i=1}^{N_{\text{gate},k}} C_i^k \quad (11)$$

$$= \kappa \sum_{\forall \text{activeEU}_k} r^k(\downarrow) \cdot \frac{\sum_{i=1}^{N_{\text{gate},k}} C_i^k}{N_{\text{gate},k}} \cdot N_{\text{gate},k} \quad (12)$$

$$= \kappa \sum_{\forall \text{activeEU}_k} r^k(\downarrow) \cdot C_{\text{avg}}^k \cdot N_{\text{gate},k} \quad (13)$$

$$\cong (\kappa \cdot C_{\text{avg}}) \sum_{\forall \text{activeEU}_k} r^k(\downarrow) \cdot N_{\text{gate},k} \quad (14)$$

$$= \varepsilon_{\text{gp}} \sum_{\forall \text{activeEU}_k} r^k(\downarrow) \cdot N_{\text{gate},k} \cdot \quad (15)$$

After considering the activity of EUs, the total power consumption can be expressed as (10) and approximated as (11) by assuming the switch activities are uniform within an execution unit; that is, $r_i^k(\downarrow) = r^k(\downarrow)$, $\forall r_i^k(\downarrow)$. Since the average output capacitances of each execution unit (C_{avg}^k) are nearly the same as the average output capacitances of total system (C_{avg}), the total power consumption can be approximated to (14). Therefore, we can obtain an approximated power estimation model shown in (15), where ε_{gp} is defined as the gate power coefficient. In this paper, we use the gate power coefficient as the unit for estimating power dissipation for the power-aware architecture.

V. RESULTS

The peak signal-to-noise ratio (PSNR) of the motion-compensated predicted frame compared to the original frame is adopted as a performance measure. Fig. 6 shows the quality degradation of the CIF video clip “table tennis” with parameters $N = 16$ and $p = 32$. The target subsample rates of GSA and ACSA are set as (4:1), (8:3), (2:1), (8:5), (4:3), (8:7), and (1:1) respectively. In the simulation, we compare our approach with GSA and the variable-search-window approach. The variable-search-window approach is simply achieved by gradually setting the search parameter p to 16, 20, 22, 25, 27, 30, and 32. As shown in the results, the quality degradation of the ACSA is less than that of the GSA and the variable-search-window approach. Fig. 7 illustrates the improvement of ACSA over GSA. Because the ACSA alleviates the aliasing problem for the



Fig. 7. Nineteenth motion-compensated frame of the “table tennis” clip. (a) GSA with the 8-to-3 subsample rate and (b) ACSA with the 8-to-3 subsample rate.

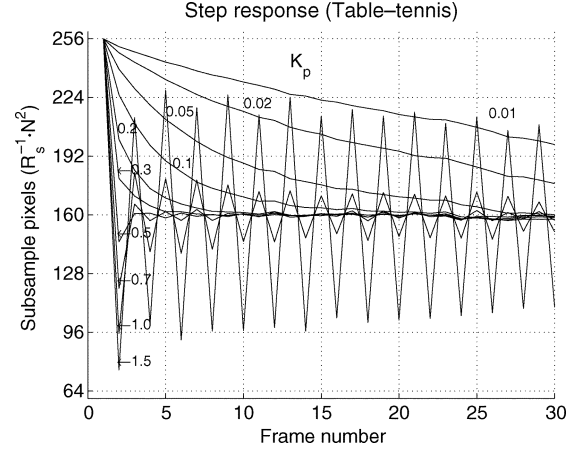


Fig. 8. Step response for varying K_p of the “table tennis” clip.

TABLE I
STATIONARY ERROR OF THE ACSA SUBSAMPLE RATE (R_{ACSA})

Target R_{ACSA}	Average R_{ACSA}	R_{ACSA} Error	Average R_{ACSA}	R_{ACSA} Error
Video Clip		Weather		
96	95.416	0.61%	95.366	0.66%
128	127.521	0.37%	127.720	0.22%
160	158.678	0.83%	159.795	0.13%
192	188.037	2.06%	191.313	0.36%
224	211.274	5.68%	221.224	1.24%
Video Clip		Table-Tennis		
96	95.398	0.63%	95.490	0.53%
128	127.467	0.42%	127.754	0.19%
160	159.533	0.29%	159.430	0.36%
192	191.429	0.30%	189.632	1.23%
224	222.893	0.49%	216.001	3.57%
Video Clip		Children		
96	95.398	0.63%	95.490	0.53%
128	127.467	0.42%	127.754	0.19%
160	159.533	0.29%	159.430	0.36%
192	191.429	0.30%	189.632	1.23%
224	222.893	0.49%	216.001	3.57%

The edge-extraction unit uses the high-pass gradient filter. The control parameter $K_p = 0.3$.

high-frequency band, the compensated frame with the ACSA has better quality than that with the GSA on the edge of the sportsman’s arm and thumb. In addition, we can find that the type of gradient filter does not make much difference to the performance of the proposed algorithm. Fig. 8 illustrates the stationary response with variable K_p . Obviously, the higher the value of K_p , the shorter the settling time and the worse the stability of the R_{ACSA} . The suitable range of K_p is from 0.1 to 0.5 after testing those four video clips. We analyzed the R_{ACSA} errors of ACSA in Table I. The average R_{ACSA} error is as low as 1.12% and the R_{ACSA} error variance is as low as 0.00024. Because the subsample rate can be nearly stationary with a given target subsample rate and power mode, the power-awareness of the proposed algorithm is very good and the ACSA can be applied for power-aware architecture.

Table II shows the synthesis result using the TSMC 1P4M 0.35- μm cell library, where the symbol R_s can be either the

TABLE II
POWER ANALYSIS OF THE POWER-AWARE ARCHITECTURE

EU_i	PEs array		SRA+PAT+MVS	EXU
	AD + Adder	Others		
N_{gate,EU_i}	117,760	58,708	46,440	17,121
$r^i(\downarrow)$	$4p^2 R_s^{-1}$	$4p^2$	$4p^2$	N^2
P_{EU_i}	$4.8 \cdot 10^8 \cdot R_s^{-1}$	$2.4 \cdot 10^8$	$1.9 \cdot 10^8$	$4.38 \cdot 10^6$
$P_{total}(\varepsilon_{gp})$	$4.8 \cdot 10^8 \cdot R_s^{-1} + 4.3 \cdot 10^8$			

$$\text{Area Overhead} = N_{gate,EXU} / (N_{gate,PEs\ array} + N_{gate,SRA+PAT+MVS})$$

$$\text{Power Overhead} = P_{EXU} / (P_{PEs\ array} + P_{SRA+PAT+MVS})$$

$$N = 16$$

$$p = 32$$

Cell library: TSMC 0.35- μm process.

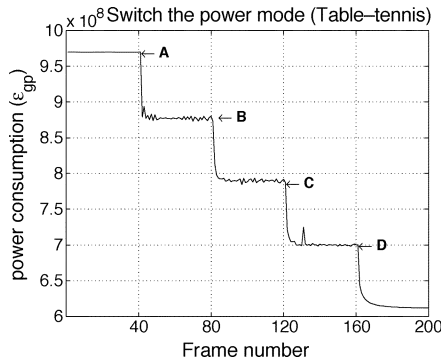


Fig. 9. Application for switching the power mode of the video clip “table-tennis.”

adaptive content-based subsample rate or the generic subsample rate, and ε_{gp} is the gate power coefficient defined in (15). Comparing with the general semisystolic architecture [4], the edge-extraction unit (EXU) of the proposed architecture is the major overhead for the power-aware function. As mentioned above, this paper uses one of three gradient filters to implement the EXU. As per the synthesis results, the gate counts of the three gradient filters are 595.33, 793.77, and 727.63, respectively. The variance of these values are very little to the overall gate count of EXU. This means that the selection of the gradient filter does not affect the overhead estimation much. Therefore, we selectively use a high-pass filter to estimate the performance overhead caused by EXU which dominates the area/power overheads. From the results, the area overhead of EXU is 7.68% and the power overheads are 0.8% and 0.5% for 4-to-1 and 1-to-1 subsample rates, respectively. Fig. 9 shows an example of the video clip “table-tennis” for switching the power consumption mode. The target subsample pixel count is reduced by 48 every 40 frames. The result shows that the adaptive control mechanism can make the power consumption reach the target level within 10 frames. According to the battery properties described in Section I, the curve shows that our power-aware architecture can match the behavior by slowly and gradually degrading the quality. The marks A, B, C, and D correspond to the power-switching points.

VI. CONCLUSION

Motivated by the concept of battery properties and the power-aware paradigm, this paper presents an architecture-level power-aware technique based on a novel adaptive content-based subsample algorithm. When the battery is in the status of full capacity, the proposed ME architecture will turn on all of the PEs to provide the best compression quality. In contrast, when the battery capacity is short for full operation, instead of exhibiting an all-or-none behavior, the proposed architecture will shift to lower power consumption mode by disabling some PEs to extend the battery lifetime with little quality degradation. As shown in the simulation results, the proposed algorithm successfully improves the compression quality of the generic subsample algorithm and does not introduce much power dissipation.

ACKNOWLEDGMENT

The authors would like to thank National Chip Implementation Center (CIC), Taiwan, R.O.C., for technical support.

REFERENCES

- [1] P. Kuhn, *Algorithms, Complexity Analysis And VLSI Architectures for MPEG-4 Motion Estimation*. Norwell, MA: Kluwer, 1999.
- [2] M. Bhardwaj, R. Min, and A.-P. Chandrakasan, “Quantifying and enhancing power awareness of VLSI systems,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 6, pp. 757–772, Dec. 2001.
- [3] D. Linden, *Handbook of Batteries*, 2nd ed. New York: McGraw-Hill, 1995.
- [4] C.-H. Hsieh and T.-P. Lin, “VLSI architecture for block-matching motion estimation algorithm,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 2, pp. 169–175, Jun. 1992.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] C. Zhu, X. Lin, and L.-P. Chau, “Hexagon-based search pattern for fast block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, May 2002.
- [9] K. Sauer and B. Schwartz, “Efficient block motion estimation using integral projections,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 5, pp. 513–518, Oct. 1996.
- [10] J.-H. Luo, C.-N. Wang, and T. Chiang, “A novel all-binary motion estimation (ABME) with optimized hardware architectures,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 8, pp. 700–712, Aug. 2002.
- [11] O.-S. Unsal and I. Koren, “System-level power-aware design techniques in real-time systems,” *Proc. IEEE*, vol. 91, no. 7, pp. 1055–1069, Jul. 2003.
- [12] B. Liu and A. Zaccarin, “New fast algorithms for the estimation of block motion vectors,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 2, pp. 148–157, Apr. 1993.
- [13] C.-K. Cheung and L.-M. Po, “Normalized partial distortion search algorithm for block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 417–422, Apr. 2000.