

RESEARCH

Open Access

Memory bandwidth-scalable motion estimation for mobile video coding

Jui-Hung Hsieh*, Wei-Cheng Tai and Tian-Sheuan Chang*

Abstract

The heavy memory access of motion estimation (ME) execution consumes significant power and could limit ME execution when the available memory bandwidth (BW) is reduced because of access congestion or changes in the dynamics of the power environment of modern mobile devices. In order to adapt to the changing BW while maintaining the rate-distortion (R-D) performance, this article proposes a novel data BW-scalable algorithm for ME with mobile multimedia chips. The available BW is modeled in a R-D sense and allocated to fit the dynamic contents. The simulation result shows 70% BW savings while keeping equivalent R-D performance compared with H.264 reference software for low-motion CIF-sized video. For high-motion sequences, the result shows our algorithm can better use the available BW to save an average bit rate of up to 13% with up to 0.1-dB PSNR increase for similar BW usage.

Keywords: motion estimation, memory bandwidth, H.264/AVC

1. Introduction

With the rapid progress of semiconductor technology, video coding is becoming popular in modern mobile devices to provide video services. In these devices, motion-compensated temporally predictive coding with motion estimation (ME) not only contributes the most to the coding efficiency of modern video encoder designs [1], but also requires large amounts of computations as well as data bandwidth (BW) [2]. This leads to severe design challenges for power-limited mobile devices. In power-limited mobile device, the available power could be changed dynamically due to low battery power or dynamic power management, such as dynamic voltage and frequency scaling [2,3]. In such cases, the available data BW could be inconsistent with the video requirements and be lower than expected. Once this situation occurs, the video coding will be delayed or forced to drop frames. Either case leads to unwanted low video quality. This BW constrained problem is getting worse with increasing camera resolution in mobile devices.

Broadly speaking, the BW-constrained ME problem is one of the resource constraints. Other resource

constrained designs [2-9] focus on lowering power consumption, with or without rate-distortion (R-D) optimization [2-5], or adjusting computational complexity with rate-control like methods [6-9]. He et al. [2] developed a new R-D analysis framework with a power constraint. Subsequently, the power-aware designs [3,4] directly change their search algorithms without R-D optimization to predesigned ones to fit a lower power mode. Chen et al. [5] used a fast algorithm and data reuse to achieve a power-aware design. Tai et al. [6] proposed a novel computation-aware scheme to determine the target amount of computation power allocated to a frame and allocated this to each block in a computation-distortion-optimized manner. The computational complexity complexity-aware designs [7-9] used a rate-control like method to combine complexity constraints into R-D optimization. The basic assumption of these approaches is that there are limited computational resources in handheld devices but sufficient memory BW. This assumption could easily fail because of dynamic mobile environment in which videos are coded and decoded at the same time or because of the dynamic power management mentioned above.

To solve the above issue, we propose a BW-scalable ME algorithm to fit the available data BW constraint. We assume that the data BW are the limited resource

* Correspondence: jhsieh.ee95g@nctu.edu.tw; tschang@g2.nctu.edu.tw
Department of Electronics Engineering & Institute of Electronics, National
Chiao-Tung University, Hsinchu, Taiwan

and could be dynamically changed [3]. The available data BW will be sufficient in full or normal battery mode and have a higher working frequency. In low battery or power-saving mode, the available data BW will be insufficient due to the lower working frequency or lower voltage supply. With a lower than expected BW supply, ME computations could fail to meet real-time constraints or lead to significant R-D performance loss due to the macroblock (MB) skipping coding. The proposed method predicts and allocates the memory BW according to its R-D gain (RDG) and the available BW to model the bandwidth-rate-distortion (B-R-D) behavior of the existing ME algorithm. This B-R-D algorithm is a rate-control like method for MB MB-based BW allocation, which maximizes the coding efficiency under the BW constraint. The simulation results show that the proposed algorithm can better utilize the BW instead of wasting it as other designs do, and it can be scaled to the available BW.

The rest of this article is organized as follows. The review of related studies is presented in Section 2. In Section 3, we propose an analytical B-R-D optimized model. The online R-D optimized BW-scalable ME scheme is summarized in Section 4. Section 5 presents the simulation results and comparisons with traditional approaches. Finally, Section 6 concludes this article.

2. Review of related studies

To solve the computational complexity and data BW challenges of ME, various approaches have been proposed, such as parallel full search hardware design and fast ME algorithms.

Full search ME designs handle the computational complexity by using parallel processing elements for matching cost computation [10]. Furthermore, with its search center at (0, 0), it can reduce the data BW by reusing the overlapped search area, termed Level C data reuse in [11]. Such a design style is simple to use, but it will need constant data BW regardless of the video contents. Besides, to meet the Level C data reuse requirement, such a design also needs a larger search range (SR) to cover the possible best matching point due to the (0, 0) search center [12], which implies a waste of data BW compared to methods with a search center at the motion vector (MV) predictor (MVP).

On the other hand, fast ME algorithms only search a few candidates so that the computational complexity is lower. To facilitate such searching, most of the fast algorithms adopt the MVP as the search center [13]. In [14], most of best matching points are around the MVP, which can cover over 90% of the best matching points within ± 8 SR. Thus, it can have a smaller SR and could have lower data BW even with poor data reuse between consecutive searches. However, even the fast ME

algorithm still assumes constant and sufficient data BW support for the required SR. Some designs with a dynamic SR [15-17] could have even lower data BW demands by changing the SR according to the content content-dependent prediction, but they still assume constant and sufficient BW support in the planning of chip design. Besides, none of the designs can adapt to dynamic data BWs. Several approaches have tried to reduce the required data BW. Designs in [18,19] use a cache to maximize the possible data reuse for irregular search patterns. Bus BW-effective ME designs in [20,21] lower the BW requirement by reducing the pixel representation from 8 bits to a binary pattern. However, these designs are only useful for specific search algorithms without a data BW constraint.

In summary, none of above approaches has considered data BW as a limited resource to explore the possibility of optimizing its usage in an R-D sense. The assumption that there will be constant and sufficient BW has the benefit of simplifying the design procedure, and thus, it is widely used in VLSI hardware design, but it usually wastes a lot of data BW because only a portion of the MBs in a high-motion video will need such a large amount of data. Such data BW waste is a serious problem for power-limited mobile devices because data access to DRAM is off-chip access and thus consumes significant power, which can be as much as the power consumption of the video chip [22]. As indicated in [22], the power consumption of external DRAM access could be up to 50% of the total power consumed by the video decoding chip. For encoding, this portion will be larger but is often neglected in the previous design. Besides, with a dynamically changing BW, the current approaches with constant and sufficient BW assumption would have insufficient BW for coding, could need more time to complete the coding and fail the real-time constraint or drop MB coding and quality to fulfill the timing constraint. Both situations are not acceptable to attain a high-quality visual experience.

3. Analytical B-R-D optimized modeling

For a given video coding distortion (or equivalent picture quality), D , and bit rate, R , if we decrease the available encoding BW, the coding will generate more distortion and bits, which in turn implies a higher D and R for ME operation and more data BW for video coding. Therefore, the overall BW usage of a ME module is linearly proportional to its search area. We introduce a set of BW control parameters, $\mathbf{B} = [\beta_1, \beta_2, \dots, \beta_L]$, to control the search area of the ME module. The model with the BW control parameters is of a more generic form and captures the available data BW under different system conditions. Consequently, the ME SR selection is then a function of these control parameters,

denoted by $SR(\beta_1, \beta_2, \dots, \beta_L)$. However, the overall BW usage of a ME module is linearly proportional to its search area. Within the BW-limited design framework, the encoder BW requirement, denoted by BW , is a function of SR , and is also a function of \mathbf{B} , denoted by

$$BW = \Phi(SR) = BW(\beta_1, \beta_2, \dots, \beta_L) \quad (1)$$

where $\Phi(\cdot)$ is the SR selection model of the ME module. To optimize the BW usage, the available data BW , β_i , should dynamically be allocated among the MBs according to their motion characteristics. Thus, we execute the ME algorithm with a different SR of BW control parameters and obtain the corresponding R-D data. According to our measurements and analysis, the R-D performance model can well be approximated by the following expression, denoted by $RDG(BW(\beta_1, \beta_2, \dots, \beta_L))$ as (2).

$$RDG(BW) = RDG(BW(\beta_1, \beta_2, \dots, \beta_L)) \quad (2)$$

where

$$RDG = RDC_{init} - RDC_{BMA} \quad (3)$$

and the RDG is the difference of the Lagrange R-D cost (RDC) at the MVP (RDC_{init}) and the final best matching position (RDC_{BMA}). The Lagrange RDC function is frequently employed as a measure of ME efficiency, which is defined as

$$RDC_{motion}(mv, \lambda_{motion}) = \min \{SAD(s, c(mv)) + \lambda_{motion}R(mv - pmv)\} \quad (4)$$

where mv is the MV received by the ME, and λ_{motion} indicates the Lagrange multiplier. The distortion term $SAD(s, c(mv))$ is the sum of the absolute differences between the original signal s and the coded video signal c . The rate term, $\lambda_{motion}R(mv - pmv)$, represents the motion information and the coded bit length of the MV difference (MVD) between the MV and predicted MV. Note that Equation 2 is computationally intensive and is intended for offline analysis to obtain the B-R-D model.

Next, we optimally configure the BW control parameters to maximize the video quality (or minimize the video distortion) and minimize the video bit rate under the BW constraint. Mathematically, this can be formulated as in (5).

$$\begin{aligned} \max_{\{\beta_1, \beta_2, \dots, \beta_L\}} \quad & RDG = RDG(BW(\beta_1, \beta_2, \dots, \beta_L)) \\ \text{s.t.} \quad & BW(\beta_1, \beta_2, \dots, \beta_L) \leq BW \end{aligned} \quad (5)$$

where BW is the available BW pool for video encoding. The optimum solution, denoted by $RDG(BW)$, describes the B-R-D behavior of the video encoder. The corresponding optimum BW control parameters are denoted by $\{\beta_i^*(BW)\}$, $1 \leq i \leq L$.

More specifically, we develop an analytical B-R-D model to perform on-line BW optimization for real-time video coding. For the simplicity of on-line execution, the RDG formulation can be well approximated by the following expression.

$$RDC_{init} - RDC_{BMA} = \gamma \times BW(\beta_1, \beta_2, \dots, \beta_L) \quad (6)$$

where γ is a positive constant. In this study, we refer to BW as the maximum required data BW for ME.

4. Online R-D optimized BW-scalable ME

Section 3 provides a theoretical analysis of the data BW-limited performance of the B-R-D optimization. However, in this section, we discuss how this theoretical limited data BW performance can be realized in practical video coding. There are four major issues that need to be addressed. First, the real BW calculation requires global knowledge of the on-chip SRAM buffer resource and reuse strategy. Second, in BW variations between video coding and decoding as discussed in this section, we assume that the available data BW for video coding are time-varying because of non-stationary video input on the real-time coding and decoding side. Third, once the optimum BW efficiency of the previous coded MB is determined, we need to develop a scheme to allocate and predict the BW interval to achieve the video smoothness constraint. This approach is computationally intensive and its corresponding parameter adjustment is only suitable for offline analysis. In real-time video encoding on mobile devices, it is desirable to develop a low-complexity scheme that is able to estimate the BW interval parameters from the frame statistics collected in the video coding. Fourth, to avoid under- or over-use of the BW pool, the target SR is further refined by the neighboring MV. In the following, we will discuss these issues.

4.1. BW budget initialization

First, the BW budget (BW_{budget}) is initialized for BW allocation of the overall data BW pool later in the coding process. This initialization takes the available system BW and converts it to a default system SR for the ME. Then, the BW budget is allocated with the above system SR for a GOP, as in (7).

$$BW_{budget} = \frac{BW_{Bus}}{Frame_Rate} \times GOP_size \quad (7)$$

where the BW_{Bus} denotes the bus data transmission rate (bytes/s), $Frame_Rate$ is the number of coded frames per second, and GOP_size denotes the frame numbers in a GOP. Larger GOP size allows for more freedom in adjusting the BW. For the purposes of having a concrete example that represents common

practices in video coding, the BW budget for the GOP is set 16 frames in this article.

4.2. BW evaluation in an R-D sense

To justify the BW usage from (6), the BW efficiency, G_{ave} , is defined as the sum of the RDG before the current coded k th MB divided by the total used BW (BW_{usage}^k), which denotes the accumulated used data BW up to the $(k - 1)$ th MB, as in (8) and (9).

$$G_{ave} = \frac{\sum_{i=1}^{k-1} (RDC_{init}^i - RDC_{BMA}^i)}{BW_{usage}^k} \quad (8)$$

where

$$BW_{usage}^k = \sum_{i=1}^{k-1} BW_{usage}^i \quad (9)$$

and RDC_{init}^i denotes the RDC at the predicted MV position. RDC_{BMA}^i denotes the RDC after the motion search of the block-matching algorithm, and BW_{usage}^k denotes the used data BW in the i th MB with a Level C data reuse scheme.

G_{ave} measures the BW efficiency by averaging the RDG over the used BW before the k th MB, which implies how much RDG can be achieved with a unit of data BW. Thus, the more G_{ave} we gain, the better BW and coding efficiency we will obtain. In the following step, we will use G_{ave} for BW prediction.

4.3. BW prediction and allocation with the smoothness constraint

With the BW efficiency, G_{ave} , we can derive the allowed BW interval with the BW prediction and allocation. The BW prediction predicts the available BW for the next coded MB with the smoothness constraint. The smoothness constraint maintains the quality and the smoothness (i.e., similar RDC) between consecutively coded MBs. With this constraint and the RDG per unit BW from (8), we can predict the forward and backward BW usage and thus, constrain the possible BW usage of the next coded MB.

First, to keep the quality and the smoothness between the current and the previous MBs, we use the RDC data from previous MBs to make further predictions (10).

$$RDC_{init}^k - G_{ave} BW_{BP}^k = \frac{\sum_{i=1}^{k-1} RDC_{BMA}^i}{k - 1} \quad (10)$$

where BW_{BP}^k denotes the backward BW prediction, as shown in latter equation. In (10), the left-hand side is the target RDC of the current MB, and the right-hand

side is the average RDC of the previous MBs. To maintain the quality and the smoothness, ideally, the target RDC of the current MB will be equal to the average past RDCs. Thus, if we have larger G_{ave} , (10) implies that less BW (i.e., BW_{BP}) is needed to maintain a similar RDG as the previous MBs. Therefore, the backward prediction for the current k th MB can be derived, as in (11) from (10).

$$BW_{BP}^k = \frac{RDC_{init}^k - \frac{\sum_{i=1}^{k-1} RDC_{BMA}^i}{k - 1}}{G_{ave}} \quad (11)$$

In contrast to BW_{BP} , we define the forward prediction BW_{FP} to keep the quality and smoothness between the current and the future MBs by adopting BW information as in (12).

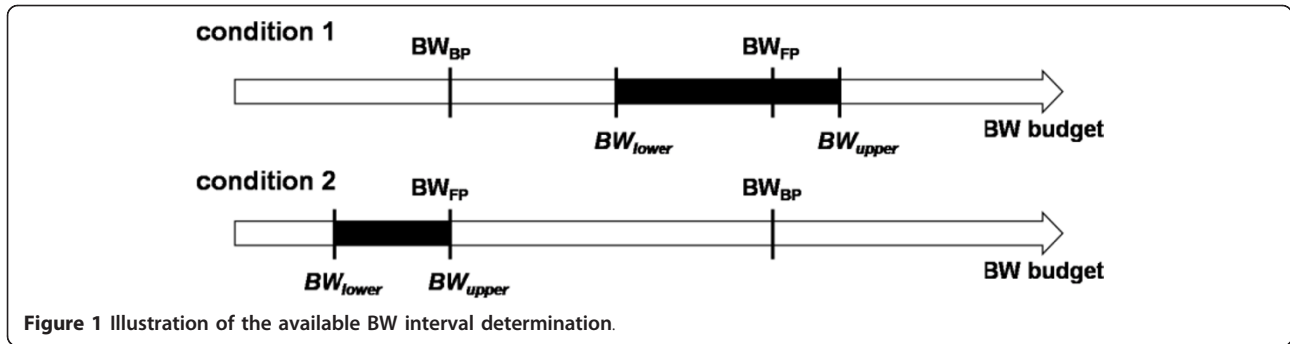
$$BW_{FP}^k = \frac{BW_{budget} - BW_{usage}^k}{n - (k - 1)} \quad (12)$$

where n is the overall MB numbers in a GOP. Because we have no knowledge of the future RDG, the forward prediction, BW_{FP} , is set to the remaining BW budget divided by the remaining MBs in the GOP that are not coded yet.

These two BW predictions link the BW usage between the past MBs and the future MBs. Their relationship can be used to allocate the available BW as follows:

$$\begin{aligned} & \text{if } (BW_{FP} > BW_{BP}) \{ \text{condition 1} \\ & \quad BW_{lower} = BW_{BP} + 0.5 \times (BW_{FP} - BW_{BP}); \\ & \quad BW_{upper} = BW_{FP} + 0.25 \times (BW_{FP} - BW_{BP}); \\ & \quad \} \\ & \text{else } \{ \text{condition 2} \\ & \quad BW_{lower} = BW_{FP} - 0.5 \times (BW_{BP} - BW_{FP}); \\ & \quad BW_{upper} = BW_{FP}; \\ & \quad \} \end{aligned}$$

in which, BW_{lower} and BW_{upper} are the lower and upper bounds of the BW usage per MB, respectively. The parameters, 0.5 and 0.25, are selected empirically and are easy to implement because they are powers of 2. The parameters are obtained from a two-step process. In the first step, we execute the proposed BW-scalable ME algorithm with different configurations of parameters to obtain the corresponding BW_{lower} , BW_{upper} , and R-D data. Note that this step is computationally intensive and is intended for offline analysis to obtain BW_{lower} , BW_{upper} , and the B-R-D model only. Once the B-R-D model and the BW intervals BW_{lower} and BW_{upper} are established, we perform the second step, which optimizes the configuration of the BW control parameters to maximize the video quality under the system



BW constraint. Meanwhile, the parameters, which are empirically selected in the following section, are obtained by the same method. For condition 1, as shown in Figure 1, BW_{BP} is smaller than BW_{FP} , which implies that less BW had been allocated to the previous MBs, and thus, more BW can be allocated to the next MB. As a result, we set the lower bound, BW_{lower} , higher than the average BW in the past MBs (equal to $BW_{BP} + 0.5 \times (BW_{FP} - BW_{BP})$), and also set the upper bound, BW_{upper} , higher than the average BW prediction in the future MB coding (equal to $BW_{FP} + 0.25 \times (BW_{FP} - BW_{BP})$). This larger BW allocation enables better quality. In contrast, for condition 2 in Figure 1, BW_{FP} is smaller than or equal to BW_{BP} , which implies that too much BW had been allocated to the previous MBs, and hence less BW can be allocated to the next MB. As a result, both bounds should be lower than BW_{FP} to keep the smoothness and quality, and we set BW_{lower} equal to $BW_{FP} - 0.5 \times (BW_{BP} - BW_{FP})$ and set BW_{upper} equal to BW_{FP} .

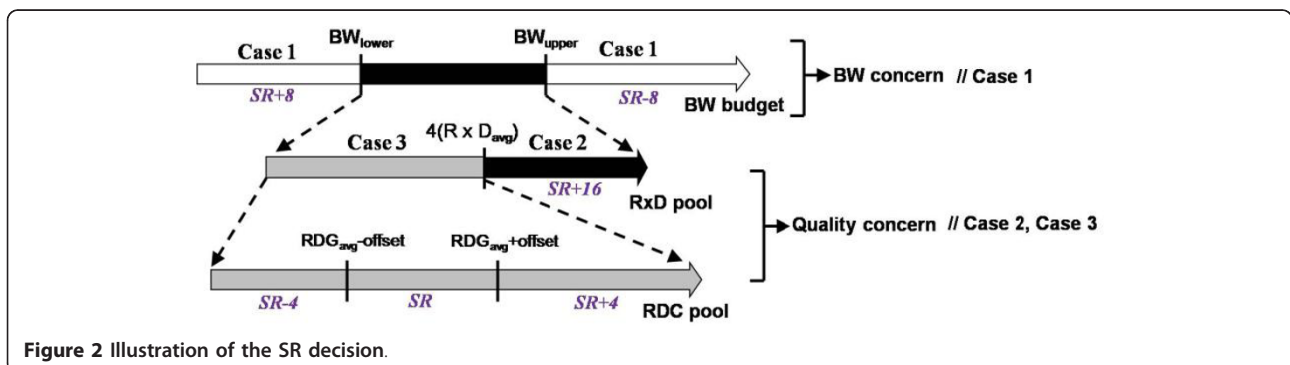
4.4. SR decision and refinement

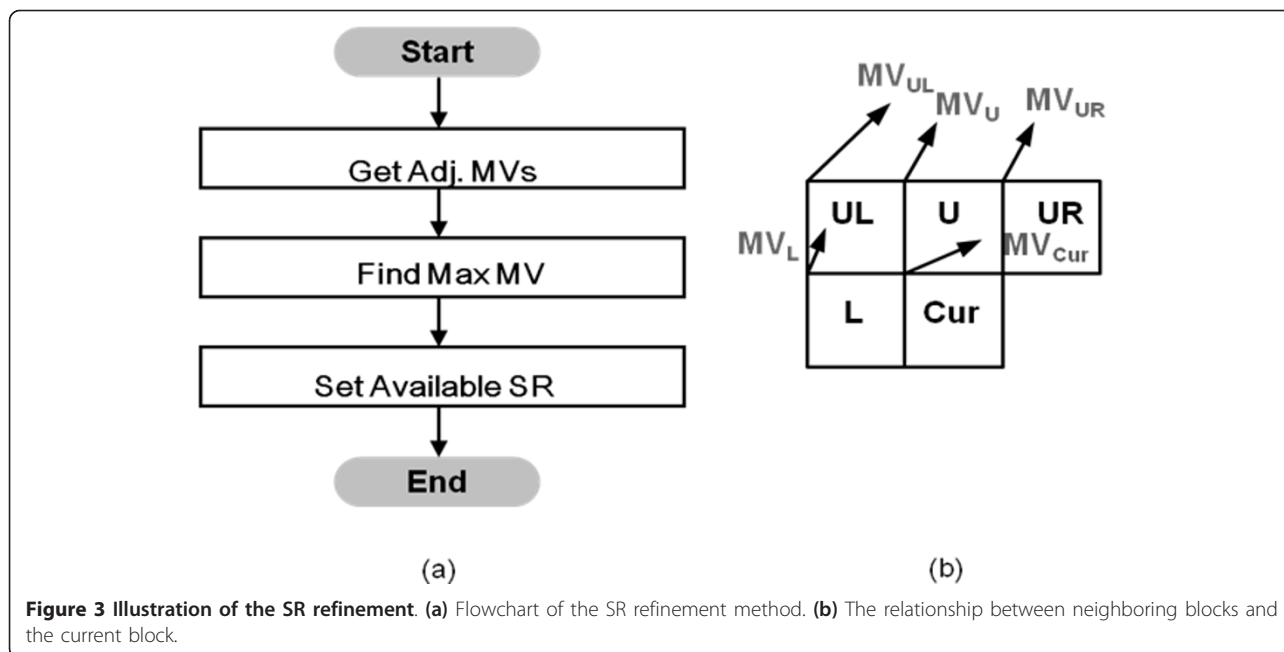
Finally, we employ the above available BW interval and R-D data to make an SR decision for the next MB coding. The SR decision is divided into three cases, and the corresponding SR adjustment coefficient is resolution independent, as shown in Figure 2. Case 1 is the BW limited case because the average BW usage of the

previous MBs falls outside the available BW interval bounded by BW_{upper} and BW_{lower} . Thus, the current SR is decreased by 8 if it is larger than BW_{upper} or increased by 8 if it is smaller than BW_{lower} for next MB coding.

The average BW usage of the previous MBs falling inside the available BW interval implies sufficient BW is available for R-D optimization. This can be further divided into two cases, case 2 and case 3. If the RDC ($R \times D_{cur}$) is larger than a predefined threshold (case 2), the video has a bad quality, and thus, the SR is increased by 16 for better quality in the next MB. This threshold is set empirically to 4 times, the average RDC of the previous MBs, i.e., $4(R \times D_{avg})$, for coarse-grained refinement of the quality. However, if the RDC ($R \times D_{cur}$) is smaller than the predefined threshold (case 3), the video has a quite smooth quality, and thus, the SR is adjusted slightly. Thus, the SR remains unchanged if the RDG of the current MB (RDG_{cur}) is within the average RDG (RDG_{avg}) plus or minus an adaptive *offset* (i.e., $RDC_{BMA}/20000$ empirically for fine-grained refinement of quality). However, if the RDG_{cur} is smaller than $RDG_{avg} - offset$, the video is of good enough quality, and thus, the SR is decreased by 4 to save BW. On the other hand, if the RDG_{cur} is larger than $RDG_{avg} + offset$, the quality is low, and the SR is increased by 4 to improve the quality.

The above SR decisions are further refined to avoid BW waste by considering the SR values in the adjacent MBs, as illustrated in Figure 3a. First, we get the





adjacent MVs from the neighboring blocks and the MV of previous frame on the co-located block, such as MV_{UL} , MV_U , MV_{UR} , MV_L , and MV_{Cur} , shown in Figure 3b. All these MVs are of sub-pel precision. Then, we compare these five MVs and choose a maximum MV (max_mv). After that, we set the available SR value using this maximum MV. The refined SR, max_avail_SR , is

$$max_avail_SR = \begin{cases} SR_{lower}, & max_mv \leq mv_{lower} \\ SR_{step} \times Ceil(\frac{max_mv}{SR_{step}}) + SR_{offset}, & mv_{lower} < max_mv \leq mv_{upper} \\ SR_{upper}, & otherwise \end{cases} \quad (13)$$

in which the parameters SR_{lower} , SR_{upper} , SR_{step} , and SR_{offset} are resolution dependent. For our simulation, we set SR_{lower} equal to 4 for CIF and 26 for HD (720P) resolution. Meanwhile, we set SR_{upper} , SR_{step} , and SR_{offset} equal to 32, 4, and 4 for CIF resolution and equal to 72, 8, and 2 for HD (720P) resolution. Meanwhile, we set mv_{lower} and mv_{upper} equal to 2 and 24 for CIF resolution and 24 and 64 for HD (720P) resolution.

Finally, the SR is selected by choosing the minimum SR between max_avail_SR and SR from Figure 2, for MB coding.

4.5. Summary of the algorithm

Figure 4 shows the proposed B-R-D optimized algorithm that can be combined with existing ME algorithms to make them BW scalable. This algorithm first models the available BW with its RDG and then predicts and allocates the BW in an R-D optimized sense to determine the available SR. The whole algorithm is repeated for all

inter-coded frames in a GOP and consists of four steps, as described below.

Step 1. **Initialization:** Create the BW budget from (7) for all MBs in a GOP.

Step 2. **BW evaluation in an R-D sense:** Evaluate the RDG in terms of the consumed BW as shown in (8) and (9) to model the BW in a R-D sense.

Step 3. **BW prediction and allocation with the smoothness constraint:** From the RDG obtained from step 2 and the available BW, the BW for the next coded MB is predicted in (10) to (12) and allocated as described in Section 4.3 to keep the video quality as smooth as possible using the smoothness constraint.

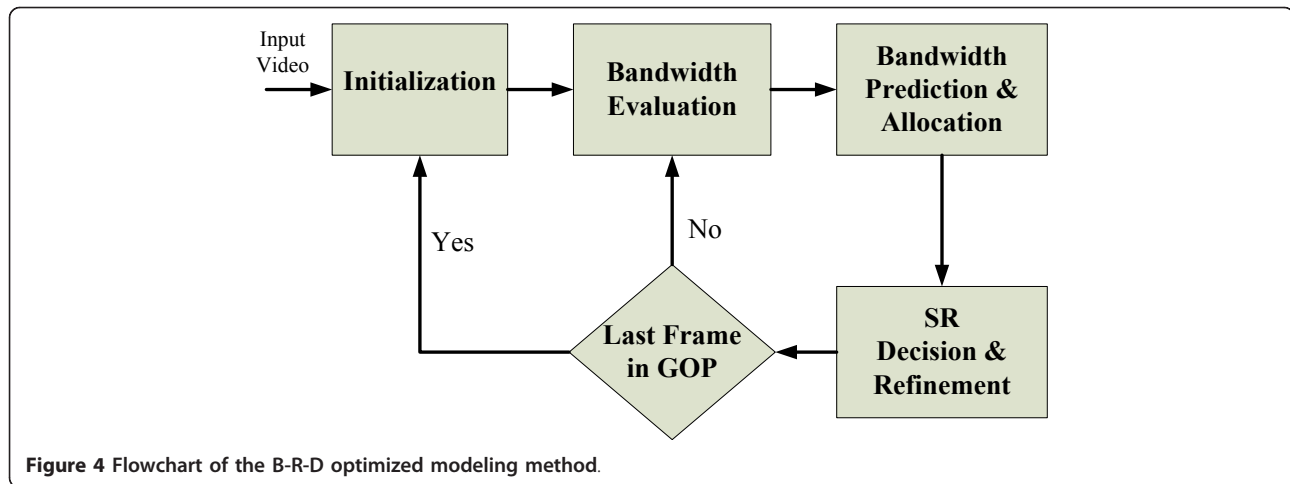
Step 4. **SR decision and refinement:** According to the available BW from step 3, the SR of next coded MB is determined and refined in (13) for ME execution.

5. Simulation results

5.1. Simulation conditions

The proposed algorithm was implemented in the H.264/AVC reference software, JM [23], for performance evaluation. The simulation conditions are CIF-sized test sequences with a baseline profile, no R-D optimization, one reference frame, a full-search algorithm as well as an Enhanced Predictive Zonal Search (EPZS) algorithm [24] for ME, IPPP sequences, 30 frames/s, and 16 frames per GOP. All of the block matching algorithms were implemented using Visual C++ on a PC with a 2.66 GHz Intel® Core™ 2 Duo CPU.

In the following simulations, we classify the corresponding BW conditions into two patterns: a constant data BW



pattern and a variable data BW pattern. Both patterns provide the same amount of reference block data for the same $SR \pm R$. However, the constant data BW pattern will assume that the available BW is constant and fixed during ME operations, which in turn assumes that the available BW is sufficient and implies that the video encoder does not have a BW constraint during the video encoding process. Meanwhile, the variable data BW pattern will assume that the available BW is variable during ME operations, which assumes that the available BW is insufficient and implies that the video encoder is BW constrained during the video encoding process. The constant data BW pattern is the scenario used in traditional ME design, which does not consider the other components, while the variable data BW pattern simulates the scenario where the BW is changing due to situations like simultaneous coding and decoding (defined as SCD mode) in a video phone or different low power modes (defined as LP mode) for mobile applications. The SCD mode assumes the decoding uses merged sequences from Stefan, Akiyo, and Football (interleaved high-motion and low-motion sequences) and sets the scene cut at a multiple of 32 frames. With the above interleaved decoded sequence, the available BW for encoding will change dynamically, as shown in Figure 5a. Figure 5b shows the LP mode with a descending trend in data BW in a power aware system. In the following simulations, we assume the SR for the search algorithm is $\pm R$ for the constant data BW pattern R and the variable data BW pattern case.

To show the benefit of the proposed scheme, we tested three different BW adaption schemes in the following simulations. The first scheme, denoted as fixed-SR, is for ME without any BW adaption scheme. Thus, the total BW for ME is equally distributed for all MB coding, and its SR setting is constant for the entire coding time. The second scheme, denoted as simple-SR, is for ME with a simple BW adaption scheme. Its BW

adaption equally distributes the available data BW to all MBs in a period, as in the fixed-SR case, but the distribution will be changed when the available BW changes. Thus, its SR adapts as well. This adaption does not consider the used BW or the related R-D information. The final scheme, denoted as BRD-SR, is the proposed B-R-D optimized BW-scalable method.

5.2. B-R-D performance evaluation

Tables 1, 2, 3, 4, and 5 show the simulation results for the constant and variable BW patterns with the different BW adaption schemes. Figure 6 shows the average BW per frame for the high-motion Stefan sequence with the quantization parameter set to 28.

For the constant BW pattern case, Table 1 illustrates that the full search ME with the proposed BRD-SR scheme can attain similar quality performance as the that with the fixed-SR scheme in the low-motion sequence (Akiyo sequence) and the medium-motion sequence (Foreman sequence), but with less BW. In case of low-motion sequence, the proposed algorithm can save 35-83% of the BW with different SRs. For the medium-motion sequence, our algorithm can save 4-45% of the BW. For the high-motion sequence (Stefan sequence), our algorithm can save an average bit-rate of up to 13% and increase the PSNR by up to 0.1 dB under the low SR constraint. Also, the simulation shows similar results as that in the full search algorithm by applying our proposed algorithm to the fast algorithm, the EPZS algorithm, which is due to our effective SR adjustment. For a fair comparison, the presented BW has considered data reuse [11] in the overlapped region between search points, and thus, only new data that are not in the local buffer will be loaded from external memory and counted in the BW usage. In summary, the proposed algorithm can save data BW for the full search and EPZS algorithms as well.

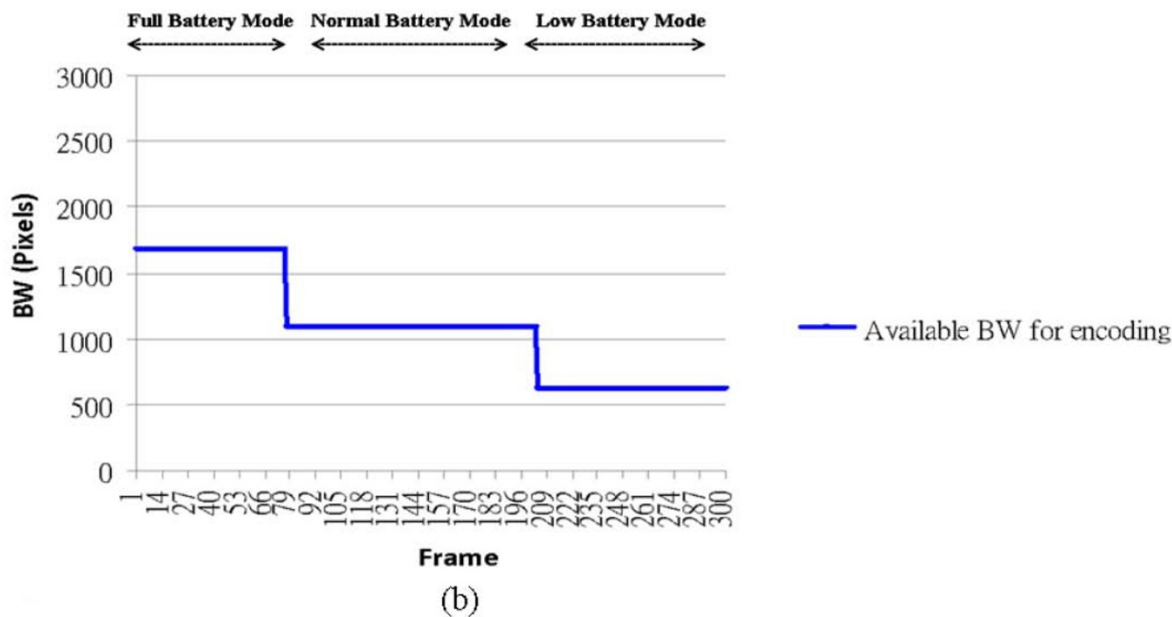
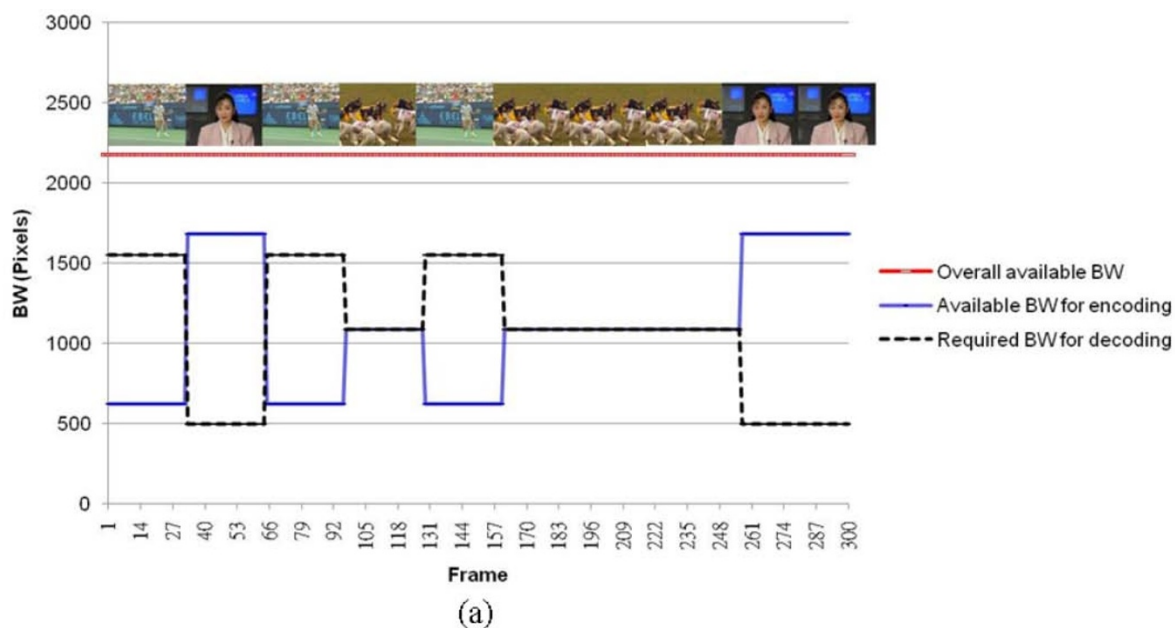


Figure 5 Variable data BW pattern with ± 8 SR for: (a) the SCD mode and (b) the LP mode.

For the variable BW pattern case, Tables 2 and 3 compare the results between the BRD-SR scheme and the simple-SR scheme in the SCD and LP modes. All of these results show trends in R-D performance and BW saving similar to those in Table 1. In summary, these results show our algorithm with B-R-D optimization can better utilize the BW for ME computation and achieves better performance than the fixed-SR and simple-SR schemes.

Table 4 shows the execution-time of the proposed algorithm and compares it to the fixed-SR scheme with the constant BW pattern. The results are similar to those found with the simple-SR scheme in the variable BW pattern case. Our proposed algorithm slightly improves execution time. However, the saving is not directly proportional to BW saving due to the calculation overhead of the MB-level BW-scalable scheme. These overheads can be reduced with further software

Table 1 Performance comparison with the fixed-SR scheme for CIF resolution

Search algorithm	Sequence	Akiyo			Foreman			Stefan		
		BW pattern	Δ BW (%)	Δ PSNR (dB)	Δ Bit-rate (%)	Δ BW (%)	Δ PSNR (dB)	Δ Bit-rate (%)	Δ BW (%)	Δ PSNR (dB)
FS	Const. 8 ^a	-35.2	-0.02	+0.24	-4.78	-0.02	+1.79	-1.01	+0.10	-13.42
	Const. 16 ^a	-69.8	-0.01	-0.35	-22.07	-0.02	+2.10	-6.04	+0.01	-2.45
	Const. 24 ^a	-82.8	-0.01	-0.45	-43.74	-0.02	+1.99	-17.59	+0.01	-1.21
EPZS	Const. 8 ^a	-31.3	-0.01	+0.07	-3.66	-0.03	+3.21	-0.25	-0.03	+2.12
	Const. 16 ^a	-65.4	-0.01	-0.17	-21.26	-0.03	+2.53	-7.14	-0.04	+3.13
	Const. 24 ^a	-79.8	+0.01	-0.45	-42.95	-0.03	+2.01	-18.75	-0.02	+1.46

^ameans constant BW and SR is set within ± 8 and ± 24 .

Table 2 Performance comparison with the simple-SR scheme for CIF resolution in the SCD mode

Search algorithm	Sequence	Akiyo			Foreman			Stefan		
		BW pattern	Δ BW (%)	Δ PSNR (dB)	Δ Bit-rate (%)	Δ BW (%)	Δ PSNR (dB)	Δ Bit-rate (%)	Δ BW (%)	Δ PSNR (dB)
FS	Variable 8 ^a	-37.8	+0.01	+0.17	-12.30	-0.02	+1.98	-1.38	+0.07	-9.83
	Variable 16 ^a	-69.9	0.00	+0.36	-31.03	-0.02	+3.19	-7.29	+0.01	-2.16
	Variable 24 ^a	-82.8	-0.01	-0.34	-45.56	-0.02	+1.69	-19.10	-0.01	-1.13
EPZS	Variable 8 ^a	-33.1	+0.02	-0.15	-11.0	-0.02	+2.64	-0.76	-0.02	+1.17
	Variable 16 ^a	-65.6	+0.01	+0.20	-29.54	-0.02	+2.37	-7.69	-0.03	+2.98
	Variable 24 ^a	-79.8	0.00	-0.09	-44.72	-0.02	+1.90	-20.8	-0.01	+1.58

^ameans variable BW and SR is set within ± 8 and ± 24

Table 3 Performance comparison with the simple-SR scheme for CIF resolution in the LP mode

Search algorithm	Sequence	Akiyo			Foreman			Stefan		
		BW pattern	Δ BW (%)	Δ PSNR (dB)	Δ Bit-rate (%)	Δ BW (%)	Δ PSNR (dB)	Δ Bit-rate (%)	Δ BW (%)	Δ PSNR (dB)
FS	Variable 8	-37.9	-0.01	+0.12	-5.05	0.00	+0.10	-3.49	+0.03	-2.83
	Variable 16	-70.2	-0.01	+0.34	-30.1	-0.02	+2.43	-16.5	+0.07	-9.29
	Variable 24	-83.0	-0.01	+0.04	-51.2	-0.02	+1.20	-32.6	-0.01	+0.04
EPZS	Variable 8	-32.9	0.00	-0.01	-3.44	-0.01	+0.37	-2.73	-0.02	+1.42
	Variable 16	-65.7	-0.01	-0.13	-27.8	-0.03	+2.84	-16.2	-0.05	+3.35
	Variable 24	-79.9	+0.01	-0.11	-49.8	-0.01	+1.49	-32.1	-0.01	+1.25

Table 4 Execution-time comparison with the fixed-SR scheme for CIF resolution

Search algorithm	Sequence	Akiyo	Foreman	Stefan
		Δ Time (%)		
FS	Const. 8	+0.45	+0.06	+0.19
	Const. 16	-0.57	-0.32	-0.06
	Const. 24	-1.94	-0.69	-0.38
EPZS	Const. 8	-1.31	-0.26	-0.45
	Const. 16	-2.31	-0.90	-0.20
	Const. 24	-3.21	-2.43	-0.90

optimization or better hardware implementation of the existing ME engine.

Table 5 shows the simulation results for the HD resolution videos and a comparison of the proposed scheme with the fixed-SR scheme. The simulation conditions are three 720P-sized video sequences with a baseline profile, no R-D optimization, one reference frame, IPPP sequences, 30 frames/s, and 16 frames per GOP. All of the simulation results show similar savings to those found with CIF resolution, which are listed in Table 1. This proves the applicability of the proposed algorithm on larger sized video sequences.

Table 5 Performance comparison with the fixed-SR scheme for 720P resolution.

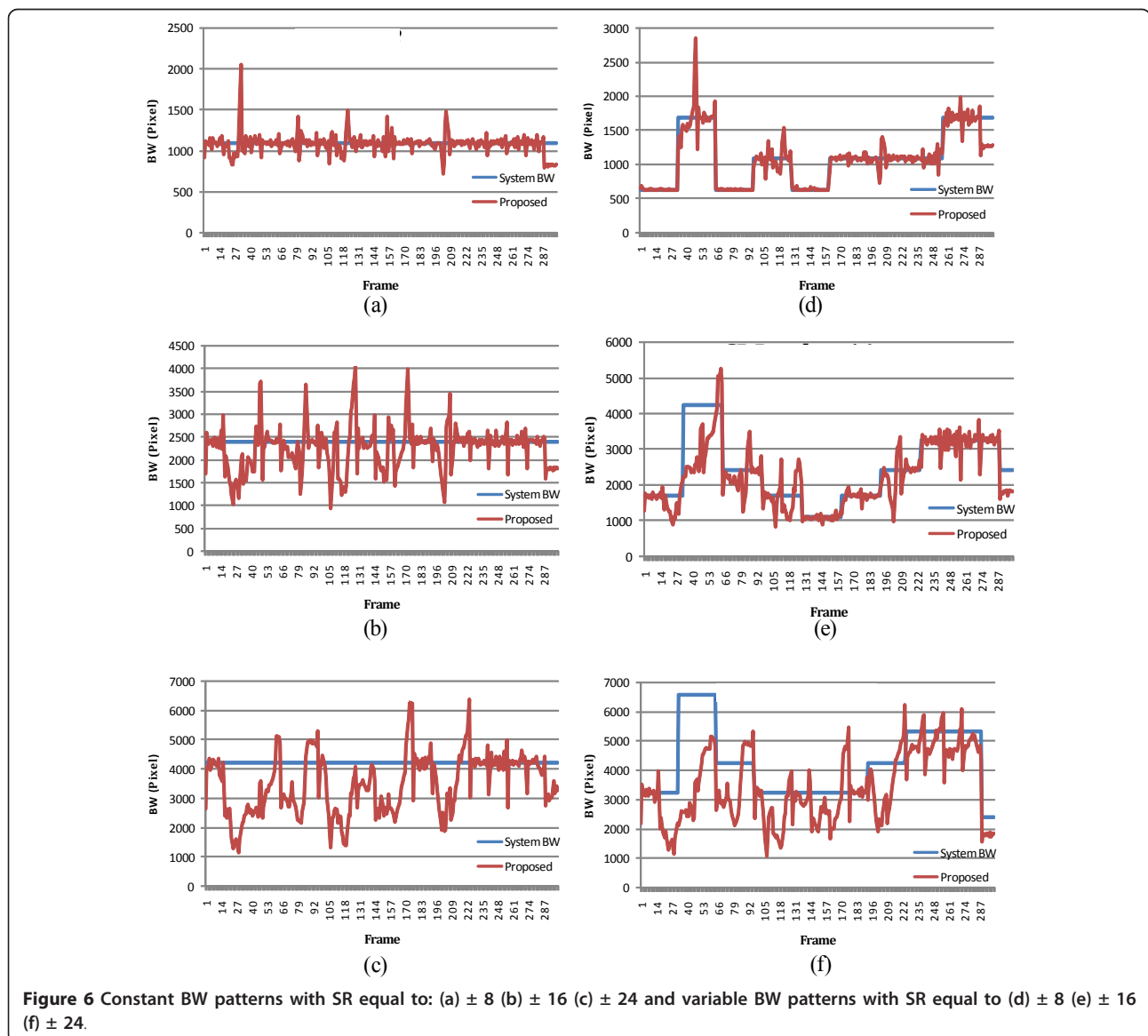
Search algorithm	Sequence	Station2			Sunflower			Tractor		
		BW pattern	Δ BW (%)	Δ PSNR (dB)	Δ Bit-rate (%)	Δ BW (%)	Δ PSNR (dB)	Δ Bit-rate (%)	Δ BW (%)	Δ PSNR (dB)
FS	Const. 56 ^a	-69.64	-0.01	+0.27	-48.98	-0.01	+0.28	-23.86	0.00	-0.11
	Const. 64 ^a	-75.97	0.00	+0.29	-59.09	-0.01	+0.20	-37.97	0.00	+0.06
EPZS	Const. 56 ^a	-69.82	-0.01	-0.06	-49.75	+0.01	-0.2	-26.52	0.00	+0.17
	Const. 64 ^a	-76.15	0.00	-0.26	-59.69	0.00	+0.39	-40.43	0.00	-0.02

^ameans variable BW and SR is set within ± 56 and ± 64 .

6. Conclusion

In this article, we propose a BW-scalable approach for an ME algorithm to maximize the R-D performance while dynamically allocating the available BW.

Compared to the traditional methods, our algorithm could save up to 70% of the BW with a full-search algorithm and 65% of the BW with the EPZS algorithm with an average SR size of ± 16 for low-motion CIF



resolution sequences. Compared to either the full search or EPZS algorithm, our proposed algorithm can save up to 70% of the BW with an SR size of ± 56 for HD (720P) resolution video. These savings come from appropriate MB-level BW allocation. In addition, while coding high-motion sequences, the simulation result shows our design could save an average bit rate of up to 13% and increase the average PSNR by up to 0.1 dB with similar BW usage for CIF resolution. The proposed design can be combined with current ME designs. Further study can be done by incorporating this work into the rate-control scheme or other resource constrained algorithms for better performance.

Abbreviations

B-R-D: bandwidth-rate-distortion; BW: bandwidth; BW_{BP} : data bandwidth backward prediction; BW_{budget} : bandwidth budget; BW_{FP} : data bandwidth forward prediction; EPZS: enhanced predictive zonal search; max_mv : maximum motion vector; MB: macroblock; MBs: macroblocks; ME: motion estimation; MV: motion vector; MVD: motion vector difference; MVP: motion vector predictor; R-D: rate-distortion; RDC: Lagrange R-D cost; RDC_{BMA} : Lagrange R-D cost at the final best matching position; RDC_{init} : Lagrange R-D cost at MVP; RDG: rate-distortion gain; SR: search range.

Acknowledgements

The authors appreciate the anonymous referees and editor for their valuable comments and suggestions that lead to the improved version of this article.

Competing interests

The authors declare that they have no competing interests.

Received: 17 March 2011 Accepted: 7 December 2011

Published: 7 December 2011

References

1. T Wiegand, GJ Sullivan, G Bjontegaard, A Luthra, Overview of the H.264/AVC video coding standard. *IEEE Trans Circ Syst Video Technol.* **13**(7), 560–575 (2003)
2. Z He, Y Liang, L Chen, I Ahmad, D Wu, Power-rate-distortion analysis for wireless video communication under energy constraints. *IEEE Trans Circ Syst Video Technol.* **15**(5), 645–658 (2005)
3. CJ Lian, SY Chien, CP Lin, PC Tseng, LG Chen, Power-aware multimedia: concepts and design perspectives. *IEEE Circ Syst Mag.* **7**(2), 26–34 (2007)
4. YH Chen, TC Chen, LG Chen, Power-scalable algorithm and reconfigurable macro-block pipelining architecture of H.264 encoder for mobile application, in *Proceedings of IEEE International Conference on Multimedia and Expo*, Ontario, Canada, pp. 281–284 (2006)
5. TC Chen, YH Chen, CY Tsai, SF Tsai, SY Chien, LG Chen, 2.8 to 67.2 mw low-power and power-aware H.264 encoder for mobile applications, *Proceedings of IEEE Symposium on VLSI Circuits*, Kyoto, Japan, pp. 222–223 (2007)
6. PL Tai, SY Huang, CT Liu, JS Wang, Computation-aware scheme for software-based block motion estimation. *IEEE Trans Circ Syst Video Technol.* **13**(9), 901–913 (2003). doi:10.1109/TCSVT.2003.816510
7. YV Ivanov, CJ Bleakley, Dynamic complexity scaling for real-time H.264/AVC video encoding, in *Proceedings of the 9th International Conference on Multimedia*, Augsburg, Germany, pp. 962–970 (2007)
8. HF Ates, Y Altunbasak, Rate-distortion and complexity optimized motion estimation for H.264 video coding. *IEEE Trans Circ Syst Video Technol.* **18**(2), 159–171 (2008)
9. CY Chang, JJ Leou, SS Kuo, HY Chen, A new computation-aware scheme for motion estimation in H.264, in *Proceedings of IEEE International Conference on Computer and Information Technology*, Sydney, Australia, pp. 561–565 (2008)
10. JF Shen, TC Wang, LG Chen, A novel low-power full-search block-matching motion estimation design for H.263+. *IEEE Trans Circ Syst Video Technol.* **11**(7), 890–897 (2001). doi:10.1109/76.931116

11. JC Tuan, TS Chang, CW Jen, On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture. *IEEE Trans Circ Syst Video Technol.* **12**(1), 61–72 (2002). doi:10.1109/76.981846
12. SS Lin, PC Tseng, LG Chen, Low-power parallel tree architecture for full search block-matching motion estimation, in *Proceedings of IEEE International Symposium on Circuits and Systems*, British Columbia, Canada, pp. 313–316 (2004)
13. P Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation* (Kluwer Academic, Norwell, MA, 1999)
14. YK Lin, CC Lin, TY Kuo, TS Chang, A hardware-efficient H.264/AVC motion-estimation design for high-definition video. *IEEE Trans Circ Syst I.* **55**(6), 1526–1535 (2008)
15. XZ Xu, Y He, Modification of dynamic search range for JVT, in *Joint Video Team, Doc JVT-Q088*, (Nice, France, 2005)
16. Z Liu, J Zhou, S Goto, T Ikenaga, Motion estimation optimization for H.264/AVC using source image edge features. *IEEE Trans Circ Syst Video Technol.* **19**(8), 1095–1107 (2009)
17. H Shim, CM Kyung, Selective search area reuse algorithm for low external memory access motion estimation. *IEEE Trans Circ Syst Video Technol.* **19**(7), 1044–1050 (2009)
18. WY Chen, LF Ding, PK Tsung, LG Chen, Algorithm and architecture design of cache system for motion estimation in high definition H.264/AVC, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, USA, pp. 2193–2196 (2008)
19. TC Chen, YH Chen, SF Tsai, SY Chien, LG Chen, Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC. *IEEE Trans Circ Syst Video Technol.* **17**(5), 568–577 (2007)
20. JH Luo, CN Wang, TH Chiang, A novel all-binary motion estimation with optimized hardware architectures. *IEEE Trans Circ Syst Video Technol.* **12**(8), 700–712 (2002). doi:10.1109/TCSVT.2002.800859
21. SH Wang, SH Tai, TH Chiang, A low-power and bandwidth-efficient motion estimation IP core design using binary search. *IEEE Trans Circ Syst Video Technol.* **19**(5), 760–765 (2009)
22. TM Liu, TA Lin, SZ Wang, WP Lee, JY Yang, KC Hou, CY Lee, A 125 μ w, fully scalable MPEG-2 and H.264/AVC video decoder for mobile applications. *IEEE J Solid-State Circ.* **42**(1), 161–169 (2007)
23. Joint Video Team Reference Software JM12.2, ITU-T <http://iphome.hhi.de/suehring/tml/download/>
24. HVC Tourapis, AM Tourapis, Fast motion estimation within the H.264 codec, in *Proceedings of IEEE International Conference on Multimedia and Expo*, Baltimore, USA, pp. 517–520 (2003)

doi:10.1186/1687-6180-2011-126

Cite this article as: Hsieh et al.: Memory bandwidth-scalable motion estimation for mobile video coding. *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:126.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com