POWER-AWARE CONTENT-ADAPTIVE H.264 VIDEO ENCODING

Avin Kumar Kannur and *Baoxin Li*

Department of Computer Engineering
Arizona State University

ABSTRACT

H.264 is a computationally intensive video codec striving for achieving the best quality for the compressed video. The computational complexity poses as a challenge for power-constrained applications. We present a system level complexity reduction for H.264 video encoding by allocating resources based on computational complexity and quality trade-off. We develop a framework which allocates the computational power of the encoder adaptive to video contents and also scales with the available battery power using a ROI classification method. Analysis is done to profile the key modules of the encoder which can be power-optimized while allocating resources. The results of the encoder module analysis are combined with the motion content analysis to obtain a power efficient encoder parameter set which reduces the computations and hence the power consumed. Our simulation results on the JM H.264 framework confirm our hypothesis and computational savings of more than 50% with quality degradation less than 1% is achieved thereby extending it's feasibility for battery powered wireless devices.

Index Terms — H.264 Video Encoding, Wireless devices, ROI coding, power optimization.

1. INTRODUCTION

Multimedia applications such as real time video capture and streaming in mobile wireless devices (e.g., video telephony) have great potentials. H.264/AVC has rich tool sets to heavily compress the video contents at desired quality and generate packetized bit streams for such wireless transmission. This efficiency comes at the cost of increased computational complexity and demanding processing power requirements. The battery technology has not progressed at the rate needed to meet the computational power requirements of multimedia rich wireless devices. In critical wireless applications such as combat mission, battery-powered video sensors need to perform power-aware processing to maximize the information conveyed. This problem can be addressed from various points of views: on the architecture level such as designing low power processor architecture, multi-cores with multiple hardware execution engines, media instruction sets [1-3]; developing ASIC to hardware-accelerate the encoder; employing multi-threaded software, and a combination of these techniques [4]. Another approach is to efficiently allocate the computational resources to the key modules of the encoder based on the perceptual relevance of the regions in the frame [5].

In this paper we present a Region of Interest (ROI) based resource allocation at the encoder on similar lines as proposed by Yang et al in [5], with a more generic content and motion adaptive power efficient encoder parameter selection framework on H.264. The paper is henceforth structured in the following manner. Section 2 covers the complexity analysis of the key functional blocks of encoder, Section 3 covers our adaptive parameter set selection process, Section 4 describes the experimental setup and simulation results and we draw conclusion from our work in Section 5.

2. COMPLEXITY ANALYSIS

In this section we first analyze the complexity of the key functional blocks of the encoder. In particular we study the non-normative part of the encoder, Motion Estimation (ME) and Rate Distortion (RD) optimization, which are the key enhancement modules incorporated in the standard, but are also heavy on processor cycle consumption. Based on the analysis, we then define a series of coding modes; each corresponding to a different level of complexity which can be used for ROI based resource allocation.

2.1. Motion Estimation Profiling

H.264 performs motion estimation at different pixel resolution (full, half and quarter pixel accuracy) and can have multiple reference frames when encoding an inter frame. The distortion metric used in ME and the search range are also some of the key elements in ME. The computational complexity of the ME process can be described as below.

$$\begin{cases} N_{Fullpel} = N_{MEmetric} * N^2 * (2*SR+1)^2 * N_{\mathrm{Ref}} \\ N_{subpel} = (N_{MEmetric} + N_{\mathrm{interp}}) * N^2 * (2*SR+1)^2 * N_{\mathrm{Ref}} \end{cases} \quad (1)$$

where $N_{Fullpel}$ and N_{subpel} are integer pel and fraction pel full search ME computations (an upper bound, in practice sub-pel search is done at best full search pel location) for a MB of size NXN, with search range SR and reference blocks N_{Ref}. $N_{MEmetric}$ *is the ME metric computation cost,* N_{interp} is the interpolation cost for a MB in fraction pel ME. In H.264, the interpolation filter uses 6 tap filter for half pel and 2 tap filter for quarter pel ME.

We analyzed the effect of the above parameter set on various test sequences. As an example the results for one test sequence (Foreman_CIF with baseline profile) are detailed below. Fig. 1 illustrates the complexity of encoding and the quality as a function of the number of reference frames. It is found that for sequences with moderate motion such as the Foreman sequence, increasing the number of reference frames will not yield significant quality improvement. However, the average encoding time per inter frame almost linearly increases with the number of reference frames, which is also expected from Eqn. (1).

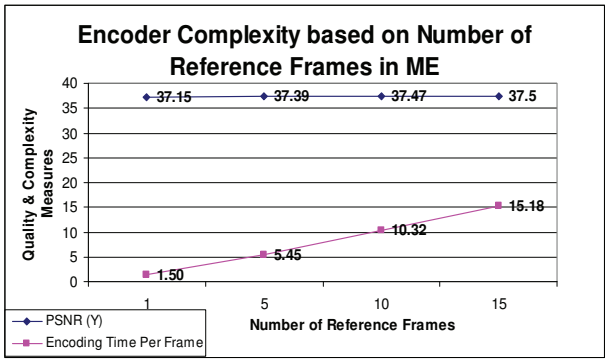


Fig 1. ME Reference frames selection complexity on a test Foreman CIF sequence (15fps, constant QP = 24). The Quality metric is PSNR Y (dB) and the Complexity metric shown is average encoding time per frame.

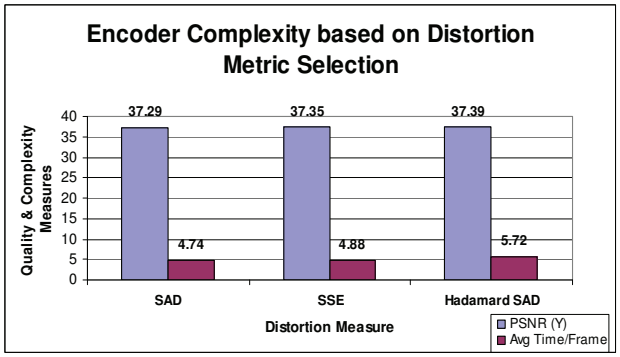


Fig 2. ME Distortion Metric Selection complexity on Foreman CIF sequence (15fps, constant QP = 24).

The distortion metric used in ME for block matching can be SAD, SSE or Hadamard SAD. We observe from Fig. 2 that Hadamard SAD yields the best matches in ME and hence the best overall quality (PSNR values). The difference in encoding time with each of the ME metrics can be captured in $N_{MEmetric}$ *(1 ADD for SAD, 1MUL & 1 ADD for SSE).* Since ME can take up to 60% of encoding time, considerable research has been done in this area and some are presented in articles [6-8].

2.2 RD Optimization Profiling

H.264 performs RD optimization with multiple coding modes for Intra and Inter frames. The mode selection algorithm computes the best possible mode (M^*) from a set of partition modes M, minimizing the overall cost using Lagrangian method using the relation given below.

$$RD_{\cos t}(M^*) = \arg\min_{M}(D + \lambda R) \quad (2)$$

This process includes ME, estimation of rate (DCT-quantization, entropy coding and Inverse quantization, IDCT) for all possible partition modes and requires high computational power. The quality of the encoded sequence depends on whether this optimization is done or not and is evident from the PSNR gains in Fig 3. We found 5-10% increase in coding complexity with RD optimization enabled. There are many research works published to expedite the mode selection process and a few are listed in [9, 10].

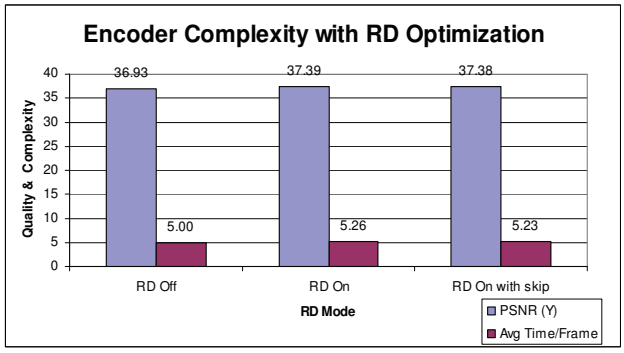


Fig 3. RD optimization complexity chart on Foreman CIF Test sequence.

We partition the modes into the following mode sets in the increasing order of complexity with ModeSet 0 being the least complex set. The grouping is made take into account the overall computation time in ME with a particular mode selected. Based on video content, the percentage of modes selected from each ModeSet varies. For example, in smooth regions ModeSet 0 will be used and in foreground regions where there are objects ModeSet 1 and 2 will be predominantly used.

$$\begin{cases} ModeSet\ \ 0 = \{INTER\ \ 16X16, SKIP\ , DIRECT\ \ \} \\ ModeSet\ \ 1 = \{INTER\ \ 16X8, INTER\ \ 8X16, INTER\ \ 8X8\} \\ ModeSet\ \ 2 = \{INTRA\ \ 16X16, INTRA\ \ 4X4, INTER\ \ 8X4, \\ INTER\ \ 4X8, INTER\ \ 4X4\} \end{cases} \quad (3)$$

3. ROI BASED ENCODER PARAMETER SELECTION

In our previous work on Region Of Interest (ROI) based rate control [11], we have proposed an algorithm that segments the video into ROI and background based on motion analysis. ROI segmentation can also be used to efficiently allocate computational resources to region of importance and hence minimize the complexities in the static background regions, as done in [5]. In the following, we exploit this idea by combining ROI-based encoding with a power scalable encoding parameter set selection process.

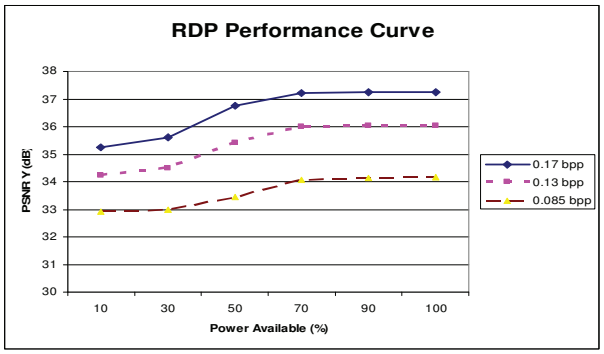
3.1 Adaptive Parameter Selection

Our ROI algorithm combines macro-blocks (MB) based on Motion Vector and Distortion into foreground and background regions and classifies them in different slice groups using the explicit mode in Flexible Macroblock Ordering (FMO). The slices so generated are independently decodable and can be

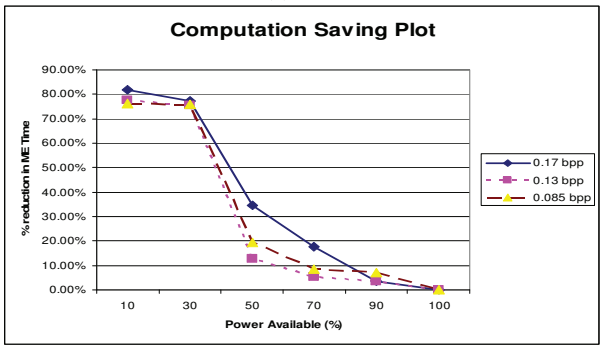
efficiently coded with ROI-based rate control [11]. Also, we can allocate more computational resources for encoding slice groups which are critical such as ROI information. The static background slice group which has little or no motion can be coded with a restrictive parameter set to speed up the encoding process. We describe two such implementations below.

3.1.1. Parameter selection based on ROI

From our analysis in Section 2, we identify the encoder parameter sets from the statistics obtained from a database of video sequences. While coding the background slice group, we pick a subset of block partition modes for RD optimization from Eqn. (3) which are less expensive computationally and exclude those coding modes which are complex from the coding decision. Since background slice is with less or no motion, excluding smaller partition modes (ModeSet 2) will not affect quality much. Further we can restrict the search range for ME to a smaller search space. The slice group with ROI is still coded with all partition modes and higher search range to maintain good perceptual quality.



(a)



(b)

Fig 4. Plot showing the performance of power scaled encoding (a) RD Power Performance curve, (b) Computational savings, for Foreman CIF sequence.

3.1.2. Power scaled parameter selection

To achieve power scalable encoding, the computational resources must be allocated based on available battery power. In addition, we should also consider the current frame complexity while allocating the resources. We use two factors, Frame Complexity factor (FC) derived from PSNR drop ratios [11] and power factor (PF) (% battery remaining) to modulate the encoder parameter set as below.

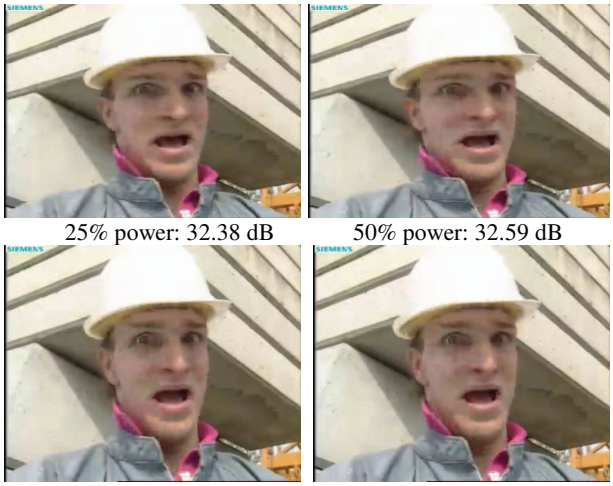
$$SR_{ROI\_Slice} = \min((SR_{Non\_ROI\_Slice} + floor((FC-1)*PF)/10),16)$$

$$N_{Ref} = ceil((FC*0.75) + PF*0.125)$$

For non ROI slice group:

$$\begin{cases} 66 < PF < 100 : \{ModeSet0,1,2\}, Full, Half\ \&\ Q\_PelME, HadamardSAD \\ 33 < PF < 66 : \{ModeSet0,1\}, Full\ \&\ Half\ \&\ Q\_PelME, SSE \\ 0 < PF < 33 : \{ModeSet0\}, FullPelME, SAD \end{cases}$$

When the battery reaches critical state (<33% remaining), we reduce the resolution of MV (for Non ROI slice) by performing ME at integer pel accuracy. The distortion measure in ROI classification will still recover the regions in the background which might potentially get degraded due to the compromise made above.



25% power: 32.38 dB 50% power: 32.59 dB

75% power: 32.96 dB 100% power: 33.04 dB

Fig 5. Subjective quality at different power levels for Foreman CIF sequence frame 93 at 128 Kbps.

4. EXPERIMENTAL RESULTS

We have conducted simulations based on the JM 12.4 H.264/AVC encoder. The encoder is setup in the baseline profile with two explicit slice grouped FMO and CAVLC entropy coding. The number of reference frames is set to 5, and the search range is 16, with RD optimization and loop filter enabled, and fast full search ME method employed. All the test video sequences are of CIF resolution with 4:2:0 sub-sampling mode and encoding rate is 15 fps.

Fig. 4 shows a plot of Rate Distortion Power (RDP) performance curve and computational savings (in terms of ME time) in power scaled encoding mode for the Foreman sequence. We observe from the plots that there is a tradeoff in quality and power savings. The subjective quality is still maintained due to our ROI based allocation process as observed from Figure 5. We conducted simulations in both Constant Bit Rate (CBR) and Variable Bit Rate (VBR) modes for several other sequences and the results are tabulated in Tables 1 and 2 respectively. The "Fixed" part is without ROI allocation and the "Adaptive" part is with ROI based allocation at 100% power. We can see that in sequences such as Hall and Foreman, where the activity is localized, the ROI classification yields the best performance which can be seen in the reduction in the encoding time (translating into computation power saving). On the contrary in

sequences such as coastguard, bus and mobile which have global and/or camera motion, the total computational saving is lower, but still significant at around 50%. The instantaneous plots of PSNR at various power levels shown in Fig. 6 and 7 confirm a smooth variation of quality in the encoded sequence.

5. CONCLUSIONS

In this paper we have presented system level encoder complexity reduction by adaptively selecting the parameter sets based on video contents and available battery power. Simulation results demonstrate power-scaled encoding efficiently allocates the computational resources with ROI quality preserved. The performance boost gained might be lower in real time video encoders which use fast ME algorithms and are developed on pipelined DSP with multithreaded software and hardware accelerators which will be a future direction to probe further on this topic. However, our approach can still be used for power-optimizing the battery-powered wireless devices for real time video capture and streaming applications.

6. REFERENCES

[1] Tung-Chien Chen , Yu-Han Chen, Chuan-Yung Tsai, Sung-Fang Tsai, Shao-Yi Chien, and Liang-Gee Chen, "2.8 to 67.2 mW low-power and power-aware H.264 encoder for mobile applications," Symposia on VLSI Technology and Circuits, p 222-3, 2007.

[2] Steven Ge, Xinmin Tian and Yen-Kuang Chen, "Efficient multithreading implementation of H.264 encoder on Intel hyper-threading architectures," Information, Communications and Signal Processing, vol 1, pages 469-473, 2003.

[3] Jui-Chin Chu, Wei-Chun Ku, Shu-Hsuan Chou, Tien-Fu Chen, and Jiun-ln Guo, "An embedded coherent-multithreading multimedia processor and its programming model," 44th ACM/IEEE Design Automation Conference, p 652-7, 2008.

[4] "Goto, K., Hatabu, A., Nishizuka, H., Matsunaga, K., Nakamura, R., Mochizuki, Y., and Miyazaki, T, H.264 video encoder implementation on a low-power DSP with low and stable computational complexity," IEEE Workshop on Signal Processing Systems Design and Implementation, p 101-6, 2006.

[5] Yang Liu , Zheng Guo Li, and Yeng Chai Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," IEEE Transactions on Circuits and Systems for Video Technology, v 18, n 1, p 134-9, Jan. 2008.

[6] Koziri, M.G, Dadaliaris, A.N., Stamoulis, G.I., and Katsavounidis, I.X, "A novel low-power motion estimation design for H.264," IEEE 18th International Conference Application-specific Systems, Architectures and Processors, p 247-52, 2007.

[7] Tung-Chien Chen, Yu-Han Chen, Chuan-Yung Tsai, and Liang-Gee Chen, "Low power and power aware fractional motion estimation of H.264/AVC for mobile applications," IEEE International Symposium on Circuits and Systems, p 4 pp, 2006.

[8] Tung-Chien Chen, Yu-Han Chen, Sung-Fang Tsai, Shao-Yi Chien, and Liang-Gee Chen, "Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC," IEEE Transactions on Circuits and Systems for Video Technology, v 17, n 5, p 568-77, May 2007.

[9] Yongsu Jo, Yong-Goo Kim, and Yungho Choi, "Fast mode decision algorithm using optimal mode predictions for H.264-based mobile devices," International Conference on Consumer Electronics, p 2 pp, 2007.

[10] Hyungjoon Kim, and Altunhasak, Y, "Low-complexity macroblock mode selection for H.264-AVC encoders," ICIP, p 765-8 Vol.2, 2004.

[11] Avin Kumar Kannur and Baoxin Li, "An Enhanced Rate Control scheme with motion assisted slice grouping for low bit rate coding in H.264", ICIP, 2008.

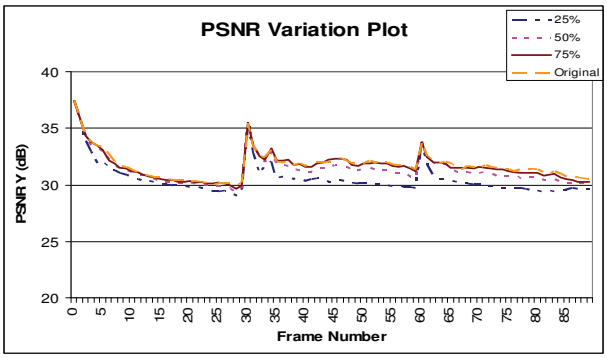


Fig 6. PSNR variation plot for Stefan CIF sequence at different power levels. Bit rate 256 Kbps, 15 fps with 3 GOPS.

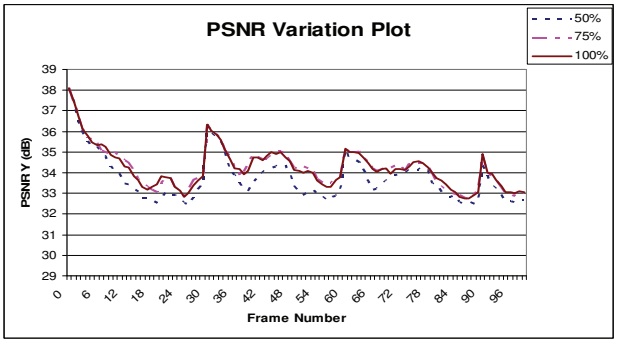


Fig 7. PSNR variation plot for Foreman CIF sequence at different power levels. Bit rate 128 Kbps, 15 fps with 3 GOPS.

Table 1. CBR simulation results. (Rate: 256 Kbps, 100 frames, 15fps)

Sequence	Encoding Time (s)		PSNR (Y) dB	
	Fixed	Adaptive	Fixed	Adaptive
Foreman	617.53	332.2	37.65	37.32
Stephan	560	420.8	31.71	31.22
Coastguard	624.55	315.55	30.43	30.21
Mobile	621.35	338.06	29.62	29.27
Hall	621.68	356.37	39.39	39.3
Bus	628.56	392.71	30.51	30.14

Table 2. Variable Bit Rate (VBR) simulation results. (QP_I=24, QP_P=28, 100 frames, 15fps, CIF 4:2:0)

Sequence	Encoding Time (s)		PSNR (Y) dB	
	Fixed	Adaptive	Fixed	Adaptive
Foreman	636	308.71	38	37.99
Stephan	641.14	359.55	36.78	36.73
Coastguard	650.06	396.83	35.79	35.76
Mobile	712.59	442.77	35.68	35.65
Hall	598.55	309.53	38.68	38.66
Bus	676.5	456.75	36.15	36.11