JOINT RATE-DISTORTION-COMPLEXITY OPTIMIZATION FOR H.264 MOTION SEARCH

Yu Hu, Qing Li, Siwei Ma and C.-C. Jay Kuo

Integrated Media Systems Center and Department of Electrical Engineering
University of Southern California, Los Angeles, California, 90089

ABSTRACT

A joint rate-distortion-complexity H.264 motion search framework is proposed to balance the encoder's coding efficiency and complexity in an embedded system environment. Under our framework, the complexity of H.264 motion search is primarily measured by the execution time of the sum of absolute differences (SAD) calculation. Two Lagrange parameters are used to terminate the complexity-inefficient motion search rounds and skip redundant motion search of small block modes, respectively. Then, the relationship between the weighted complexity and the Lagrange parameters is explored to allocate the complexity cost among different coding units. It is demonstrated by experimental results that the proposed method can reduce the complexity without much sacrifice in coding efficiency.

1. INTRODUCTION

Motion estimation is one of the most time-consuming units in the H.264 encoder due to the use of multiple reference frames, variable block sizes and fractional pixel interpolation. Several fast motion search algorithms such as the diamond search and the hexagon search were developed to accomplish a significantly faster speed while maintaining similar R-D performance as compared to the full search. They achieve a great success in the GPP system. However, the speed-up of these fast motion search algorithms when implemented on embedded systems is however not as impressive due to the limited resource of the embedded environment. Thus, it is critical to study an effective tradeoff between resource utilization in an embedded system and coding efficiency.

The complexity-constrained motion estimation has been studied for a long while. For example, Kossentini *et al.* explored the complexity-constrained MPEG-2 motion search in [1], where little attention was paid to the bit rate increase due to lower computational complexity. There were also efforts to reduce the motion compensation cost in the decoder. Some researchers studied the decoder-friendly encoder algorithm that generates low-decoding-complexity and high-quality bit stream for decoders [2]. The complexity reduction in the H.264 encoder in an embedded environment is quite different from previous work on complexity reduction in general purpose processor (GPP) systems, and most previous research was conducted from a pure algorithmic viewpoint. To our best knowledge, there has been little research conducted to balance the R-D performance and the complexity cost of H.264 motion estimation in the encoder end under an embedded environment.

Our proposed scheme is motivated by two observations. First, for fast motion search algorithms such as the diamond search, not every round of local refinement can achieve equally good SAD reduction. By eliminating the complexity-inefficient SAD operations, motion estimation can be accelerated at the cost of little bit rate increase. Second, due to the video signal characteristics, motion estimation of smaller block modes is often redundant. For instance, less than 10% MBs are encoded by block modes smaller than 8×8. Thus, by skipping motion search of unnecessary modes, we can speed up the motion estimation process without sacrificing the coding efficiency much. Our overall objective in this work is to minimize the complexity of the motion search yet maintaining high video quality.

Simply speaking, we propose a joint rate-distortion-complexity (R-D-C) optimization framework to balance the coding efficiency and the complexity cost of the H.264 encoder in this work. The method can cut off the complexity-inefficient motion search rounds, skip redundant motion search of small block modes and terminate motion search at the optimal R-D-C points. Our scheme saves the complexity for the motion search up to 35% with a small bit rate increase and negligible video quality degradation.

2. PROPOSED MOTION SEARCH SCHEME

2.1. Rate-Distortion Optimization

In video coding, a rate control algorithm dynamically adjusts encoder parameters as as to achieve a target bit rate and video quality, where the Lagrangian method is a widely accepted approach for bit allocation. The Lagrangian method is applied into two stages: motion compensation and residue coding. In the stage of motion compensation, specifically, for each block B with fixed block mode M, the motion vector associated with the block is selected through a joint rate-distortion (RD) cost function [3] via

$$J^{R,D}_{Motion} = D_{DFD} + \lambda_{Motion} R_{Motion}, \tag{1}$$

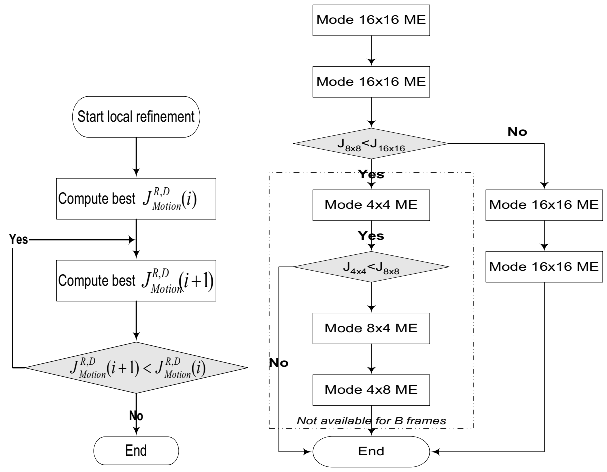
where R_{Motion} is the estimated bit rate to record the motion vector, D_{DFD} is the prediction error between the current and the reference blocks and $J^{R,D}_{Motion}$ is the joint R-D cost comprising of R_{Motion} and D_{DFD}, and λ_{Motion} is the Lagrange multiplier to control the weight of the bit rate cost. $J^{R,D}_{Motion}$ is widely used to determine the optimal displacement vector. The process is shown in Fig. 1(a), where $J^{R,D}_{Motion}$ is employed to terminate any further search effort in the fast motion search schemes and $J^{R,D}_{Motion}(i)$ is the best joint R-D cost in the i*th* search round.

Similarly, the joint cost of distortion and block mode selection in the residual coding stage can be written as

$$J^{R,D}_{Mode} = D_{Rec} + \lambda_{Mode} R_{Rec}, \tag{2}$$

where R_{Rec} is the estimated bit rate associated with mode M. D_{Rec} is the difference between the reconstructed MB and the reference

The research has been funded by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.



(a) Local refinement (b) Fast mode decision

Fig. 1: Local refinement and mode decision with $J^{R,D}_{Motion}$.

one, and λ_{Mode} is the Lagrange multiplier. The motion vectors associated with the optimal block mode will be the coded data recorded in the bit stream.

The Lagrange multipliers used in the above two cost functions determine the relative weights between the signal quality and the bit rate. To simplify the search procedure, an empirically derived relationship as shown below is generally used in practice if SAD is used in modeling D_{DFD} while SSD is used for D_{Rec}:

$$\lambda_{Motion} = \sqrt{\lambda_{Mode}}. \tag{3}$$

2.2. Rate-Distortion-Complexity Optimization

Two new Lagrange parameters "λ'_{Motion}" and "λ'_{Mode}" are adopted in our R-D-C framework to control the tradeoff between the R-D feature and complexity consumption. Parameter "λ'_{Motion}" is used to determine the motion search process in one block mode while parameter "λ'_{Mode}" is employed to decide whether the motion search of subsequent block modes (of smaller block sizes) should be conducted or not.

We can include the complexity cost in the R-D optimization cost in (1) via

$$J^{R,D,C}_{Motion} = J^{R,D}_{Motion} + \lambda'_{Motion} C_{Motion}, \tag{4}$$

where $J^{R,D}_{Motion}$ is the R-D cost function defined in (1), C_{Motion} is the complexity cost function for given block type B and mode M, λ'_{Motion} is the Lagrange multiplier for the complexity term, and $J^{R,D,C}_{Motion}$ is the newly defined joint R-D-C cost function in our scheme, which replaces $J^{R,D}_{Motion}$ in Fig. 1(a). By using $J^{R,D,C}_{Motion}$, the motion search process of a specific block mode will be terminated at the point where the R-D joint cost function $J^{R,D}_{Motion}$ reduction is not worth the complexity cost. Due to the low-efficiency of 7-block-mode search for H.264, some simplified algorithms have been developed to eliminate unnecessary modes. In [4], instead of testing all block modes from the largest to the smallest blocks, this fast algorithm tests mode 8×8 after mode 16×16 as shown in Fig. 1(b). If the combined R-D cost of mode 8×8 is less than mode 16×16, the search jumps directly to mode 4×4 while ignoring modes 16×8 and 8×16. By doing so, unnecessary modes are skipped to speed up the search process. A similar idea can be used to skip modes 8×4 and 4×8.

Table 1: Block modes and their SAD complexity.

Index	Block Mode	IC	ET (cycle)	Weight$_{IC}$	Weight$_{ET}$
1	16×16	1180	631	15	13
2	16×8	634	337	8	7
3	8×16	662	363	8	8
4	8×8	325	183	4	4
5	8×4	152	88	2	2
6	4×8	168	95	2	2
7	4×4	80	49	1	1

Due to the large number of capacity misses occurring in the switch of modes, the complexity cost should include the decision whether it is worthwhile to continue to search in subsequent modes. Thus, instead of using $J^{R,D}_{16\times16}$ and $J^{R,D}_{8\times8}$ in the comparison modules for the simplified algorithm [4] as shown by grey rhombuses in Fig. 1(b), we define a new cost function as

$$J^{R,D,C}_{Mode} = J^{R,D}_{Motion} + \lambda'_{Mode} C_{Mode}, \tag{5}$$

where $J^{R,D}_{Motion}$ is the R-D cost function given in (1), C_{Mode} is the complexity due to search using block modes of 8×8 or 4×4, λ'_{Mode} is the Lagrange multiplier, and $J^{R,D,C}_{Mode}$ is the newly defined R-D-C cost function, which is used in the simplified algorithm to decide whether motion search should be continued in subsequent modes or not. That is, once the reduction of joint R-D cost function $J^{R,D}_{Motion}$ is not worthwhile, the price paid by motion search for mode 8×8 or 4×4, its subsequent block modes is saved to accelerate the motion estimation process.

For simplification, we adopt the restriction given in Eq. (3) to limit the search space for the complexity Lagrange parameters. In the experimental section, *i.e.*, Sec. 3, the following relationship between λ'_{Motion} and λ'_{Mode} will be employed

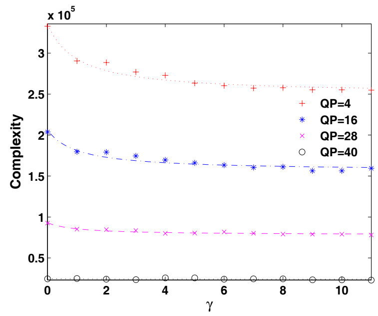
$$\lambda'_{Mode} = \sqrt{\lambda'_{Motion}}. \tag{6}$$

For the joint R-D-C function discussed earlier, we need a quantitative model for the complexity associated with each candidate motion vector and the block mode. Generally speaking, there are two types of complexity measurement: instruction count (IC) based and execution-time (ET) based. The former employs the instruction count as a metric to determine the weight of SAD operations under different block modes. This method works for the situation where cache miss does not play an important role in program execution, *e.g.*, a GPP system or an embedded system with a reasonably large cache. An even simplified estimation can be given by considering the block size alone. For example, if the SAD cost of one 4 block is one unit, then the SAD cost of one 16×16 block is 16 units. The ET-based method is more accurate since it uses the profiling data to decide the weight assigned to each mode. The instruction-based and ET-based results are listed in Table 1 for comparison.

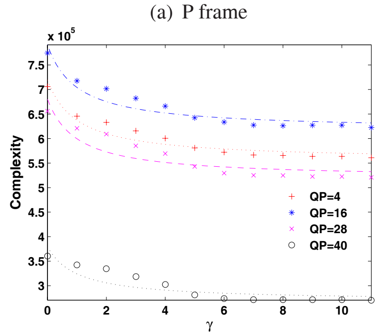
Then, the complexity cost of the ith block mode motion search can be expressed by

$$\mathrm{C}_i = N_i \times W_i, \tag{7}$$

where N_i is the SAD number of block mode i ($1 \le i \le 7$), and W_i is the associated complexity weight. The ET-based weight given in Table 1 is used in our experiment. Please also note that the complexity weight is for block modes $i = 4$ or 7 only (*i.e.* 8×8 and 4×4 blocks) in the computation of C_{Mode}. The complexity of a video frame is the weighted complexity sum of all the SAD calculations required by this frame.



(a) P frame



(b) B frame

Fig. 2: Relationship between the Lagrange multiplier and average complexity per frame.

2.3. Complexity Control

Complexity control, which is analogous to rate control, is a procedure to determine parameters such as Lagrange multipliers λ'_{Motion} and λ'_{Mode} so as to allocate complexity costs among different coding units. In this section, we will describe complexity modeling and multiplier selection which is used to characterize the relationship between the target complexity and the Lagrange multiplier.

For complexity control, an important task is to determine the relation between the complexity and the control parameter such as the Lagrange multiplier. There is little theoretical analysis available for H.264 motion search. Consequently, we resort to simulations. The relationship between the average complexity per frame and parameter γ, which is related to the Lagrange multiplier λ'_{Mode} via (9) is shown in Fig. 2 for P and B frames. Experimental data points are indicated by various symbols while the fitting curves are polynomial functions in Fig. 2. Thus, we can derive a complexity model as

$$C = a_0(t) + \frac{a_1(t)}{\gamma + 1}, \tag{8}$$

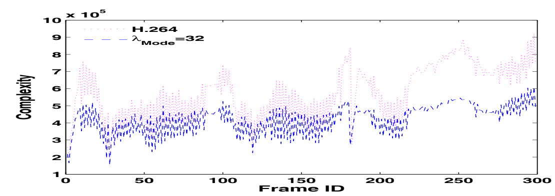
where C is the weighted average complexity of each frame, $a_0(t)$ and $a_1(t)$ are model parameters to be learned during the coding procedure. and

$$\gamma = \begin{cases} log_2(\lambda'^2_{Mode}) + 1, & \lambda'_{Mode} \geq 1, \\ 0, & \lambda'_{Mode} = 0. \end{cases} \tag{9}$$

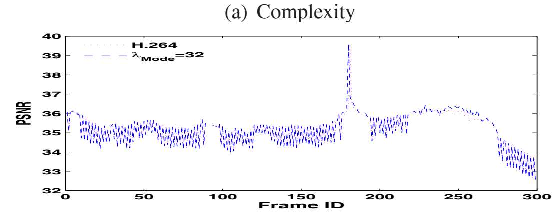
Therefore, by the adjustment of the Lagrange parameters in our proposed method, the weighted complexity can be efficiently reduced. Due to a different coding mechanism, P and B frames have different model parameters and they have to be handled separately.

Table 2: Description of Experimental Data and Parameters.

Sequence information	
Sequence name	Akiyo, Foreman, Mobile and Stefan
Frame size	CIF (352 $\times$ 288 pixels)
Video format	30fps, GOP 300, IPBPB sequence
Simulation Parameters	
QP	1, 4, 10, 16, 22, 28, 34, 40, 46 and 51
λ'_{Motion}	0 to 1024
H.264 Encoder	
Block mode	All on
Motion estimation	Diamond
S-P Frame	No



(a) Complexity



(b) PSNR

Fig. 4: Frame-to-frame computational complexity and video quality comparison.

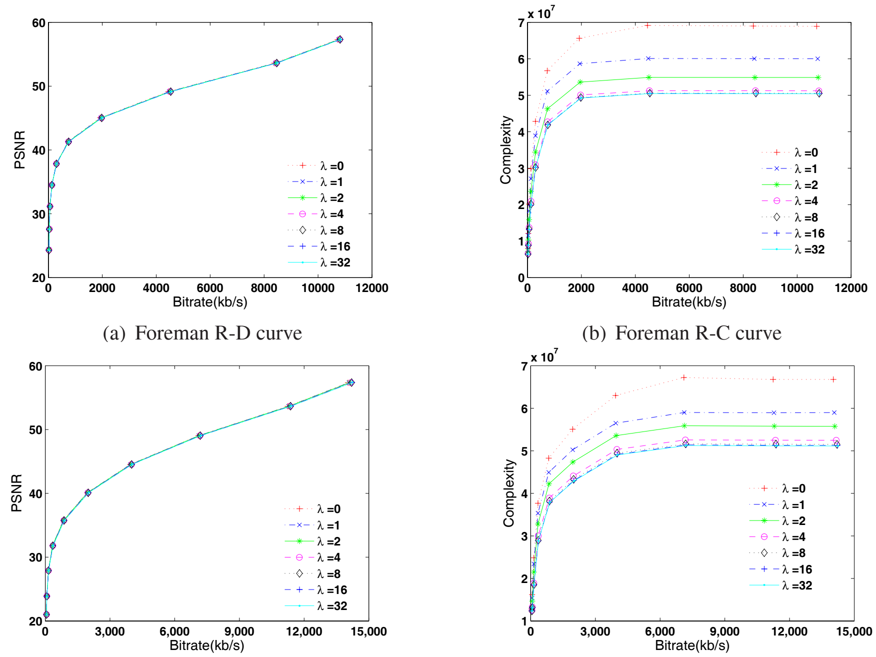
3. EXPERIMENTAL RESULTS

The experiment environment used in our simulation, including the test data and chosen parameters, is listed in Table 2. Four typical test sequences, representing the low, medium, high and very high motion conditions, are chosen. Due to the speed and code size concern , x.264 [5] is used as our H.264 reference codec.

Due to the limited space, only the results of ”Foreman” and ”Stefan” sequences are exhibited here. Fig. 3 shows the rate-distortion (R-D) curves and the rate-complexity (R-C) curves parameterized by different λ'_{Mode} values. The complexity value is measured in terms of the accumulated weighted SAD complexity required by 100 frames in the motion search process. We have several important observations. First, by choosing different values of λ'_{Mode}, it is efficient to control the computation complexity. Up to 35% of the SAD search cost can be eliminated within a small range of λ'_{Mode}. Second, the R-D performance is well maintained while the complexity decreases. As shown in these figures, the PSNR degradation is less than 0.2dB and the bit rate increase less than 3% when compared to the original bitstream with $\lambda'_{Mode} = 0$.

Furthermore, the frame-to-frame complexity and quality comparison over the entire ”Stefan” sequence with QP equal to 28 are shown in Fig. 4(a) and Fig. 4(b), respectively. We see a saving of up to 26.2% complexity cost with less than 2% bit rate increase and 0.02 dB quality loss.

The R-D-C performance at different QP values for $\lambda'_{Mode} = 32$ is summarized in Table 3, where we show the difference in the PSNR degradation (PD) between the bit stream generated by the original H.264 encoder and our complexity control algorithm. The results



(a) Foreman R-D curve (b) Foreman R-C curve

(c) Stefan R-D curve (d) Stefan R-C curve

Fig. 3: The rate-distortion (R-D) and rate-complexity (R-C) curves for "Foreman" and "Stefan" under different Lagrange multipliers.

Table 3: R-D-C performance.

		Quantization Parameter			
		4	16	28	40
Akiyo	PD (10^{-2}dB)	0.033	0.029	0.024	0.415
	CS (%)	6.58	5.43	7.54	7.15
	BI (%)	0.78	1.58	0.33	0.98
Foreman	PD(10^{-2}dB)	-1.625	1.73	10.2	0.064
	CS (%)	27.0	25.1	29.4	34.2
	BI (%)	0.90	2.39	0	2.24
Mobile	PD (10^{-2}dB)	0.56	2.82	4.07	3.32
	CS(%)	13.7	12.0	9.76	14.2
	BI (%)	.865	1.41	1.39	0.576
Stefan	PD (10^{-2}dB)	13.75	7.37	1.30	0
	CS (%)	26.7	25.9	26.2	26.0
	BI (%)	0.424	0.655	1.65	2.40

confirm that a large amount of complexity saving (CS) is achieved at the cost of a small bit-rate increase (BI) and almost zero quality degradation. Complexity saving for different test sequences can vary depending on the content type such as motion activity. There is little motion in the "Akiyo" sequence so that its motion search cost is low. For the "Mobile" sequence, there is steady camera motion (slowly panning to the left) so that it is challenging. Although every MB is moving, yet the motion vector for each MB is small due to the steady slow motion. Even for this challenging case, our proposed method can still achieve about 15% complexity saving while keeping the R-D characteristics. The excellent performance of the proposed algorithm can be explained below. During the local refinement process in motion search, not every round of search can achieve an equal effect in SAD reduction. Removing those less efficient ones for complexity saving only increases the prediction error slightly. Thus, the bit rate goes up slightly while the PSNR value is maintained.

4. CONCLUSION

A joint rate-distortion-complexity motion estimation was proposed for H.264 motion search in this work. The complexity of H.264 motion search is measured by the execution time of SAD calculation, which is a function of the block size. Specifically two Lagrange parameters are used to cut off the complexity-inefficient motion search rounds and skip redundant motion search of small block modes respectively. Then, the relationship between the weighted complexity and the Lagrange parameters is explored to determine the complexity costs in different coding units. A wide spectrum of test sequences with low to high motion was chosen to demonstrate the strength of our proposed complexity-adaptive motion search algorithm. Up to 35% of motion search complexity can be saved at the encoder with less than $0.2dB$ PSNR loss and a maximum increase of 3% bit rate.

5. REFERENCES

[1] Gallant, M.; Cote, G.;Kossentini, Faouzi, "An efficient computation-constrained block-based motion estimation algorithm for low bit rate video coding," *Image Processing, IEEE Transactions on*, vol. 8, no. 12, pp. 1816–1823, Dec. 1999.

[2] Yong Wang, Shih-Fu Chang, "Complexity adaptive motion estimation and mode decision (camed) in low power h.264," http://www.ee.columbia.edu/dvmm/researchProjects/PervasiveMedia/CAMED/camed summary.html.

[3] Sullivan, G.J.; Wiegand, T., "Rate-distortion optimization for video compression," *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 74 – 90, Nov 1998.

[4] Zhou, Z.; Sun, M.-T., "Fast macroblock inter mode decision and motion estimation for H.264/MPEG-4 AVC," in *Image Processing, International Conference on*, vol. 2, pp. 789 –792.

[5] Videolan, "x264," http://downloads.videolan.org/pub/videolan/vlc/0.8.4/contrib/.