

Complexity Modeling of Spatial and Temporal Compensations in H.264/AVC Decoding

Szu-Wei Lee and C.-C. Jay Kuo

Ming Hsieh Department of Electrical Engineering and Signal and Image Processing Institute
University of Southern California, Los Angeles, CA 90089-2564, USA
E-mails: suzweile@usc.edu and cckuo@sipi.usc.edu

Abstract—Complexity modeling of spatial-temporal compensations in H.264/AVC decoding is performed by examining a rich set of inter- and intra-prediction modes. Specifically, we study the relationship between motion vectors, frame sizes and properties of reference frames since they are related to cache management efficiency. The proposed models provide good estimation results for test video bit streams over a wide range of bit rates. As an application, an H.264/AVC encoder equipped with these models estimates the decoding complexity and chooses the best inter- or intra-prediction mode to meet the decoding complexity constraint of a target decoding platform. The decoding complexity of the resultant bit stream is reduced at the cost of small PSNR loss.

Index Terms — H.264/AVC, motion compensation, decoding complexity, rate-distortion optimization (RDO)

I. INTRODUCTION

H.264/AVC [1] is an emerging video coding standard proposed by ITU-T and ISO/IEC. Since it offers various inter- and intra-prediction modes to improve the coding gain, its decoding complexity is about 2.1 to 2.9 times more than the H.263 decoder [2]. To save the power of H.264/AVC decoding in mobile applications, one solution is to generate H.264/AVC decoder-friendly bit streams. That is, the encoder generates a bit stream that is easy to decode in a typical decoding platform (*i.e.*, with a lower decoding complexity). This motivates our current research in complexity modeling of the H.264/AVC decoder. Since a decoder consists of many components, a general model with all components included is too complicated to address in a single paper. Instead, we focus on complexity models for inter- and intra-predictions in this work, which correspond to the motion compensation process (MCP) and the spatial compensation process (SCP), respectively.

The MCP complexity of H.264/AVC was simply modeled as a function of the number of interpolation filters in [3], which in turn depends on the inter-prediction mode and the motion vector (MV) resolution. Intuitively, an MB with fewer interpolation filters should have lower decoding complexity. This model is however not accurate for two reasons. First, H.264/AVC provides a rich set of inter- and intra-prediction modes that cannot be well handled by this model. Second, this model does not take the relationship of MVs and frame sizes into account, which have an impact on the cache management efficiency and play an important role in CPU performance and decoding complexity. More recently, an MCP complexity model that considers the cache management issue was proposed in [4], [5].

There are three major contributions in this work. First, we refine the MCP complexity model furthermore by taking multiple reference frames into account, since the distribution

of selected reference frames affects cache management efficiency. Second, the SCP complexity model is proposed. The combination of these two models provides better estimation results for bit streams over a wide range of bit rates. Finally, the application of MCP and SCP models to H.264/AVC decoding complexity reduction is examined.

II. REVIEW OF MCP COMPLEXITY MODEL

In [4], the MCP complexity is modeled as

$$C_{mcp} = \alpha \cdot N_c + \beta \cdot N_y + \gamma \cdot N_x + \mu \cdot N_v, \quad (1)$$

where N_c is the number of cache misses, N_y is the number of y-direction interpolation filters, N_x is the number of x-direction interpolation filters, N_v is the number of MVs per MB, and α , β , γ and μ are weights. The number of MVs per MB is determined by the inter-prediction mode. Interpolation filters are needed when MV is of sub-pel accuracy. This factor was also considered in [3]. Since the decoder may have different implementations of interpolation filters along the x- and y-directions, two different terms are used in (1). The numbers of the x- or y-direction interpolation filters are determined by the inter-prediction mode and the MV resolution (*i.e.*, int- or sub-pel).

The number of cache misses in (1) is included to take the spatial relationship between consecutive MVs into account. When two consecutive MVs point to two reference blocks which are closer to each other, most data required for the second block tend to be in the CPU cache or internal registers after the decoding of the first block so that the number of cache misses is lower. In contrast, if two consecutive MVs point to two reference blocks which are far apart, the number of cache misses is higher. The method to count the number of cache misses is summarized below. A cache model, which contains 64 entries and each entry has 8 bytes, is used to count the number of cache misses for 4x8 and 4x4 blocks. The cache examines the address and length of each memory access during the MCP operation. The number of cache misses is added by one if the required data of MCP operations cannot be found in the cache. The address of memory access is determined by MV and the reference frame index while the length of memory access is decided by the inter-prediction mode and the MV resolution [4]. The number of cache misses for an $M \times N$ block rather than 4x8 and 4x4 is simply the row number (*i.e.*, M) or the row number plus five (*i.e.*, $M+5$) if the y-direction MVs are of sub-pel accuracy.

The weights in Eq. (1) can be obtained as follows. First, several pre-encoded bit streams are selected and Intel Vtune

performance analyzer is used to measure the number of clock ticks spent by MCP operations. Second, N_c , N_x , N_y and N_v are counted individually for those pre-encoded bit streams. Finally, the constrained-least-squared method is used to find the best fitting of weights, *i.e.*, α , β , γ and μ .

The relationship between the MCP complexity and the frame size was also examined in [5], where the decoding platform is assumed to have two-level cache management. Since the size of the first-level (L1) cache is small, only a few data of reference frame can be cached. In contrast, the second-level (L2) cache is larger and it might be able to store the whole reference frame. The L2 cache hit rate decreases as the frame size becomes large. Since the L1 cache miss penalty highly depends on the L2 cache hit rate [6], the frame size can affect the L1 cache miss penalty and, in turn, weight α in the MCP complexity model. Thus, weight α is a function of the frame size and should be trained accordingly.

III. ENHANCED MCP COMPLEXITY MODEL WITH MULTIPLE REFERENCE FRAMES

In this section, we consider an enhanced MCP complexity model that allows multiple reference frame motion estimation (MRF-ME) in the H.264/AVC encoder. As assumed in the last section, the decoding platform has two-level cache management.

We argue that the distribution of selected reference frames should play a role in the MCP complexity model, too. Typically, the first reference frame is frequently selected as the best reference frame during MRF-ME while others are rarely used. Under such a scenario, the number of L2 cache misses could be small since the first reference frame can be stored in the L2 cache. However, if all reference frames are uniformly selected as the best reference frame, the number of L2 cache misses becomes larger. Thus, the distribution of selected reference frames can affect the L2 cache hit rate. As mentioned before, the L1 cache miss penalty (or α in our model) highly depends on the L2 cache hit rate [6]. This implies that the distribution of selected reference frames affects the L1 cache miss penalty (or α) and, therefore, the MCP complexity.

To verify the above conjecture, two experiments were conducted on the Pentium 1.7 GHz mobile CPU platform. We modified the H.264/AVC encoder such that only one inter-prediction (or intra-prediction) mode was selected to code whole MBs. All bit streams are with {IPPP.P} picture structure and coded by int-pel MRF-ME mode, where the maximal number of reference frames is five. In addition, Intel Vtune performance analyzer 8.0 was used to measure the MCP complexity for all bit streams.

In the first experiment, Susie and Football sequences of the D1 format (720x480) were selected as test sequences, and the modified H.264/AVC encoder encoded whole MBs of P pictures with the P16x16 mode. Different bit streams were generated under different bit rate constraints so that the distribution of selected reference frames may vary. MCP complexities (in terms of decoding time in milli-seconds) for

these bit streams are shown in Table I. We see that MCP complexities of the Football sequence for 64Kbps and 9.6Mbps are about the same. The distributions of the five reference frames are also quite similar. In contrast, MCP complexities of the Susie sequence vary from 43.43ms to 64.16ms as its bit rate moves from 64K to 9.6M. The distribution of the five reference frames also changes. It is interesting to compare the MCP complexity of the Susie bit stream of 2.56 Mbps and that of Football bit streams. They are similar to each other (about 50.7-52.7ms) while distributions of selected reference frames are very similar, too.

TABLE I
MCP COMPLEXITIES (DECODING TIME IN MILLI-SECONDS) FOR BIT STREAMS WITH DIFFERENT DISTRIBUTIONS OF SELECTED REFERENCE FRAMES (RFs), WHERE SUSIE (S) AND FOOTBALL (F) SEQUENCES ARE ENCODED UNDER DIFFERENT BIT RATE CONSTRAINTS.

Bit stream	Time (ms)	1st RF	2nd RF	3rd RF	4th RF	5th RF
S 9.6M	64.16	60.14%	16.22%	10.56%	6.45%	6.63%
S 2.56M	50.68	76.84%	10.96%	6.52%	2.89%	2.80%
S 256K	47.07	83.57%	8.50%	5.02%	1.43%	1.48%
S 64K	43.43	89.14%	5.81%	3.91%	0.53%	0.61%
F 9.6M	51.92	77.47%	9.79%	5.50%	3.56%	3.68%
F 64K	52.67	77.67%	9.37%	6.63%	2.76%	3.57%

Since the bit streams are all coded by the P16x16 mode with integer-pel MV resolution, the two decoding complexity terms (*i.e.*, N_x and N_y) in Eq. (1) are zero. We may infer that the distribution of selected reference frames affects either the weight for the number of cache misses (*i.e.*, α) or that for the number of MVs (*i.e.*, μ). This is further clarified with the Susie D1 as the test sequence.

In the second experiment, we consider two groups of bit streams, where distributions of selected reference frames are similar within the same group but different between two groups. Bit streams with similar distributions of selected reference frames were generated by the modified H.264/AVC encoder with the same quantization parameter (QP). The modified H.264/AVC encoder generated 7 bit streams encoded by the P16x16, P16x8, P8x16, P8x8, P8x4, P4x8 and P4x4 inter-prediction modes, respectively. Since these 7 bit streams were generated from the same video sequence and QP, it is likely that the same reference frame is selected by the MRF-ME process to predict an MB and, as a result, they have similar distributions of selected reference frames. Each group of bit streams was used to train the weight for the number of cache misses and that for the number of MVs individually. We observe that these two groups have different weights for the number of cache misses but similar weights for the number of MVs.

The above two experiments clearly demonstrate that the distribution of selected reference frames affects the L1 cache miss penalty and, thus, weight α in the MCP complexity model. Weight α should be determined by taking the frame size and the distribution of selected reference frames into account. For more details on selection of α , we refer to Sec. V-A.

IV. SCP COMPLEXITY MODEL

To model the SCP complexity, several experiments were conducted on the Windows XP Pentium 1.7 GHz mobile CPU platform. The modified H.264/AVC encoder was used to encode Foreman CIF (352x288), Football D1 and Blue sky HD (1920x1080) sequences, where whole MBs of these three video sequences were encoded by the I8MB DC prediction direction (PD) and chroma DC PD. Then, Intel VTune 8.0 was used to measure the decoding complexities (decoding time in milli-seconds) of these three test bit streams. It is observed that the decoding complexities of these three different frame sizes of bit streams are quite similar, where the decoding complexities per MB for chroma DC PD and I8MB DC PD are about $7.7 \cdot 10^{-5}$ ms and $4.1 \cdot 10^{-4}$ ms, respectively.

These results imply that cache miss might have a little impact to the SCP complexity for two reasons. First, the SCP operation can be treated as a 2D memory access from neighboring MBs to the current MB. It is likely that neighboring MBs are already in the CPU cache so that the cache miss probability is very low when the SCP operation is performed. Second, the SCP complexities of bit streams coded by the same intra type and PD but with different frame sizes are quite similar. Since the weight for the number of cache misses varies for different frame sizes as mentioned before, this implies that the number of cache misses should be small for SCP. More experiments for different intra types and PDs were conducted to verify this conjecture furthermore.

Since the cache miss has less impact to the SCP complexity, the decoding complexities of those bit streams coded by the same intra type and PD are similar. Consequently, the SCP complexity is simply modeled as a function of the number of PDs for a specific intra type here. It is written mathematically as

$$C_{scp} = \sum_{i=0}^3 N_{16,i} \cdot \omega_{16,i} + \sum_{i=0}^8 N_{8,i} \cdot \omega_{8,i} + \sum_{i=0}^8 N_{4,i} \cdot \omega_{4,i} + \sum_{i=0}^3 N_{c,i} \cdot \omega_{c,i}, \quad (2)$$

where $N_{16,i}$, $N_{8,i}$, $N_{4,i}$ and $N_{c,i}$ are the numbers of PDs for the I16MB, I8MB, I4MB intra types and the MB chrominance component, respectively, and $\omega_{16,i}$, $\omega_{8,i}$, $\omega_{4,i}$ and $\omega_{c,i}$ are weights (*i.e.*, the decoding complexity per MB of the corresponding PD). For more details on the selection of these weights, we refer to Sec. V-B.

V. EXPERIMENTAL RESULTS

We conducted experiments to verify the enhanced MCP and SCP complexity models given in (1) and (2) on the PC platform. The CPU was Pentium mobile 1.7 GHz CPU with 512 Mb RAM and the operating system was Windows XP. The reference JM9.4 decoder was optimized by the Intel MMX technology.

A. Selection of Weight α for MCP

1) *Training Phase:* Weight α for the number of cache misses in MCP was obtained by the following steps. First, the modified H.264/AVC encoder was used to generate several bit streams coded by different QPs varying from small to large (*i.e.*, from high to low bit rates), where the P16x16 mode and int-pel MRF-ME were used to code whole MBs. Bit streams coded by different QPs may have variant distributions of selected reference frames. For example, since MVs with larger reference indices require more bits, it is likely that the first reference frame is selected as the best reference frame in low bit rate by the MRF-ME process.

Once the MCP complexities of these bit streams are measured by Intel Vtune, the weight for the number of MVs decided in [4] was used to determine weights α for the number of cache misses for these bit streams. In our work, Foreman CIF, Susie D1, and Blue sky HD were selected to train weights α for the number of cache misses. Each CIF, D1 and HD bit streams contain 270, 68 and 14 frames, respectively, and were encoded by the modified H.264/AVC encoder with different QPs, (*i.e.*, QP=10, 12, ..., 34).

2) *Weight Selection Phase:* First, the distribution of selected reference frames in coded video for recent 128 MBs is collected. To estimate the MCP complexity of an MB, the best weight α for the number of cache misses is selected from a set of weights trained by the above steps according to the frame size and the distribution of selected reference frames. Since the distribution of selected reference frames is a monotonically decreasing function as given in Table I, we use the distance of two entropy functions to measure the similarity between two distributions of selected reference frames. Then, we select weight α whose distribution of selected reference frames is most similar to that of the recent 128 MBs of underlying video as the desired weight. For the remaining three weights (*i.e.*, β , γ and μ) of the MCP complexity model, the trained results in [4] were used in this work.

B. Selection of Weights $\omega_{x,i}$ for SCP

To obtain weights of the SCP decoding complexity model, we used Foreman and Mobile CIF sequences as training sequences. Since there are four, nine and nine PDs for the I16MB, I8MB and I4MB intra types, respectively, the total number of PDs, which can be used to code the MB luminance component, is 22. The modified H.264/AVC encoder generated 22 bit streams for all PDs while the MB chrominance component was coded by the DC PD only. Besides, the modified H.264/AVC encoder generated 6 bit streams for the remaining three chrominance PDs, *i.e.*, horizontal, vertical and plane directions, where three of these six bit streams whose MB luminance components were coded by the I8MB DC PD while the MB luminance components of the remaining three bit streams were coded by I4MB DC PD. As a result, there were 28 bit streams used to train weights of the SCP complexity model in each training sequence.

Note that not all PDs can be used to code whole MBs. For example, since the vertical PD cannot be used to code those

MBs in the upper frame boundary because those reference MBs are not available, the modified H.264/AVC encoder chose the DC PD to code those MBs within the upper frame boundary. Thus, bit streams were coded by at most two PDs with the same intra type. Intel Vtune performance analyzer 8.0 was used to measure the decoding complexities for all bit streams. The weight of the DC PD is first obtained via dividing the measured decoding complexity of the bit stream with DC PD only by the number of MBs. As a result, once the weight of the DC PD is determined, the weights of other PDs can be obtained easily, too. The weights of chroma PDs can be determined by the similar approach above.

C. Accuracy of Joint MCP/SCP Complexity Model

The comparison between the proposed joint MCP/SCP complexity model and the actual decoding complexity measured by the Intel Vtune is shown in Table II. We see that the proposed joint MCP/SCP complexity model can provide fairly accurate estimation results for bit streams over a wide range of bit rates. Errors between the actual and estimated decoding complexities as listed in the 3rd column are all within 6%. The PSNR values of the corresponding bit streams are given in the last column.

TABLE II

COMPARISON BETWEEN ACTUAL DECODING COMPLEXITY (AC) AND ESTIMATED DECODING COMPLEXITY (EC) FOR CIF, D1 AND HD BIT STREAMS (DECODING TIME IN MILLI-SECONDS)

Bit stream	AC	EC	Error	PSNR (dB)
Flower CIF 2.048M	145.81	137.66	5.58%	39.19
Foreman CIF 1.536M	183.10	179.31	2.07%	44.78
Stefen CIF 512K	147.05	143.89	2.15%	34.77
Tempete CIF 256K	137.23	141.92	3.42%	33.26
Akiyo CIF 128K	44.36	41.79	5.80%	44.14
Container CIF 64K	36.75	36.82	0.19%	38.45
Susie D1 7.680M	756.70	715.88	5.39%	46.42
Ship D1 5.120M	344.36	325.53	5.47%	42.04
Football D1 2.560M	324.37	325.21	0.26%	36.95
Blue HD 29.40M	433.15	434.65	0.35%	47.32
Toy HD 15.36M	542.62	537.45	0.95%	40.68
Sunflower HD 2.56M	377.44	399.62	5.88%	41.72

D. Application to Complexity Reduction

Experimental results that the H.264/AVC encoder equipped with the joint MCP/SCP complexity model and the decoding complexity control scheme proposed in [5] generates bit streams for decoding complexity reduction are shown in Table III. Errors between actual and target decoding complexities are all less than 7%. The resultant bit streams can save a significant amount of decoding complexity at the cost of some PSNR loss. This is particularly interesting in a mobile broadcasting environment, where multiple mobile devices will get broadcast/streaming video in real time.

As compared with sequences without decoding complexity control as given in Table II, the bit stream of CIF Foreman with the target decoding complexity at 130 ms loses 0.43 dB in PSNR (from 44.78dB in Table II to 44.35dB in Table III) but saves 30.04% in decoding complexity. The D1 Football bit stream with the target decoding complexity at 220 ms loses

0.22 dB in PSNR (from 36.95dB in Table II to 36.73dB in Table III) but saves 29.35% in decoding complexity. Finally, the HD Sunflower bit stream with the target decoding complexity at 290 ms loses 0.2 dB in PSNR (from 41.72dB in Table II to 41.52dB in Table III) but saves 26.33% in decoding complexity. It is possible to trade the complexity saving for the quality of coded video. For example, we can increase the complexity of the CIF Foreman from 130ms to 150ms and raise the PSNR value from 44.35dB to 44.57dB. Similar cases are provided for Football and Sunflower sequences in Table III.

TABLE III

DECODING COMPLEXITY CONTROL FOR FOREMAN CIF 1.536M, FOOTBALL D1 2.560M AND SUNFLOWER HD 2.560M BIT STREAMS, WHERE AC IS THE ACTUAL DECODING COMPLEXITY MEASURED BY

INTEL VTUNE

Bit stream	AC	Error	PSNR (dB)	Saving
Foreman (150ms)	148.75	0.84%	44.57	18.76%
Foreman (130ms)	128.09	1.49%	44.35	30.04%
Football (260ms)	264.48	1.70%	36.84	18.46%
Football (220ms)	229.16	4.00%	36.73	29.35%
Sunflower (330ms)	310.77	6.19%	41.63	17.66%
Sunflower (290ms)	278.07	4.29%	41.52	26.33%

VI. CONCLUSION

Complexity models of temporal and spatial compensations in H.264/AVC decoding were presented and their application to decoding complexity reduction was examined in this work. The joint MCP/SCP model helps the encoder select proper motion vector, inter- and intra-prediction modes, and then generate a video bit stream that is most suitable for a receiver platform with some hardware constraint. These models were shown to provide fairly good estimation results for bit streams over a wide range of bit rates. An H.264/AVC encoder equipped with the joint MCP/SCP model can generate bit streams to meet different decoding complexity constraints. The resultant bit streams can be decoded at a lower complexity at the cost of small PSNR degradation.

REFERENCES

- [1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC coding standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 7:560–576, July 2003.
- [2] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro. H.264/AVC baseline profile decoder complexity analysis. *IEEE Trans. on Circuits and Systems for Video Technology*, 7:704–716, July 2003.
- [3] Y. Wang and S. F. Chang. Complexity adaptive H.264 encoding for light weight stream. In *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pages II25–28, May 2006.
- [4] S. W. Lee and C.-C. Jay Kuo. Complexity modeling for motion compensation in H.264/AVC decoder. In *IEEE Int. Conf. on Image Processing (ICIP2007)*, Sep. 2007.
- [5] S. W. Lee and C.-C. Jay Kuo. Motion compensation complexity model for decoder-friendly H.264 system design. In *IEEE Int. Workshop on Multimedia Signal Processing (MMSP2007)*, Oct. 2007.
- [6] J. Hennessy and D. A. Patterson. Computer architecture: A quantitative approach 2nd edition. In *Morgan Kaufmann*, page 417, 1996.