

Efficient Constrained Video Coding for Low Complexity Decoding

Paulo J. Cordeiro^{1,2,3}, Juan Gomez-Pulido², and Pedro A. Assunção^{1,3}

¹ Inst. Politécnico de Leiria - ESTG
Campus 2, Morro do Lena, Leiria, Portugal
cordeiro@estg.ipleiria.pt

² Universidad de Extremadura, Dept. Tecnología de Computadores y
Comunicaciones, 10071 Cáceres, España

³ Instituto de Telecomunicações Campus 2, Morro do Lena, Leiria, Portugal

Abstract. In the latest video compression standard H.264/AVC motion compensation is the most complex operation mainly because of the operations involved on computing sub-pixel predictions. Generally this increases a video stream decoding complexity. This paper describes an efficient method to generate compressed video streams with low complexity decoding requirements. This is particularly targeted at portable decoders, where the use of such video streams leads to reduced power consumption extending the battery life. By using metrics of the computational complexity needed to decode a video stream and make the coding process aware of those, it is shown that the decoding complexity can be significantly reduced with minimal punishment in rate-distortion performance. The experimental results indicate that such optimized constrained coding method is able of achieving substantial decoding complexity decrease at the expense of minimal PSNR loss within the scope of operational bit rates.

1 Introduction

The explosive growth in wireless and wired networking technology has profoundly changed the global communications infrastructure. It is the convergence of wireless, multimedia, and networking that is fundamentally changing the way people communicate with each other. Today, video coding lies at the core of multimedia. Current portable devices such as mobile phones, personal digital assistants, palmtops, etc are increasingly used to access, decode and render multimedia content in which compressed video plays a major role. The limitation imposed by battery life and computational constraints of processing equipment have been a strong motivation for research on complexity issues of video coding and decoding systems in the last few years [1,2,3]. Distinct approaches can be done to pursuit the final goal of extending the time that multimedia content can be delivered and consumed in portable equipment, result that is desirable by both, consumer and service providers. Reduction of the computational complexity [4] and power saving mechanisms [5] are among the most popular approaches to deal with this issue.

Although power saving in portable devices is dependent on the characteristics of several different functional components of the device itself such as hardware, operating system, processing software implementation, communication protocols, etc, in video decoders this is highly related to the computational complexity of the decoding process. It is particularly relevant in the case of H.264/AVC video coding standard where complexity, at both the encoding and decoding sides, is major concern in any practical real time implementation. Besides all implementation optimizations that can be considered for the purpose of minimizing decoding complexity, there is still room for further reduction by producing compressed video streams such that their inherent decoding complexity is made lower [6,7]. This can be done by constraining the encoder in order to reduce the use of the most complex coding tools and options which lead to higher decoding computational complexity. The challenge is how to achieve this goal without compromising too much the signal quality when compared with unconstrained encoding.

Motion compensation has determinant impact on decoding complexity of H.264/AVC streams due to sub-pixel computation process which demands for a great processing effort. This is because computation of predictions from sub-pixel motion vectors is done by interpolation filters which need a different amount of filtering operations according to the location of the sub-pixel to be determined. In this paper we propose possible metrics to quantify the decoding computational complexity based on the amount of filtering operations needed to compute half and quarter-pixels, not only for rate-distortion but also to decoding complexity in selecting the best motion vector for each block process. Normal rate-distortion performance is then compared with the created rate-distortion-complexity model which in turn is also evaluated.

In the remaining sections a brief review of relevant issues of the H.264/AVC standard is given on next section, with particular emphasis on motion compensation and half/quarter-pixel interpolation, also the proposed rate-distortion-complexity optimization procedure is described in section 3, and the experimental results are presented and discussed in section 4.

2 Motion Compensation and Sub-pixel Interpolation in H.264/AVC

As in previous Moving Picture Experts Group (MPEG) / International Telecommunication Union Telecommunication Standardization Sector (ITU-T) video coding standards, the H.264/AVC video codec defines the video signal structure as a series of groups of pictures (GOPs) comprising frames of luminance (Y) and two chrominance (Cb and Cr) pixels. Each individual picture is organized into slices which in turn are divided into non-overlapping blocks i.e. macroblocks (MBs). Slices are typically encoded as either Intra (I), Predictive (P), or Bi-Predictive (B) where I slices are encoded using predictive pixels from the same frame and are independent of all others, P slices use inter-frame prediction methods as well as intra-prediction methods and B slices use an expanded

set of forward and/or backward inter-prediction methods compared to P-frames. For P and B slices, motion estimation (ME) is carried out for each MB to find the best match from a reference frame. In H.264/AVC, each MB can be further partitioned into sub-blocks and multiple reference frames can also be used. An integer transform (IT), with similar characteristics to those of the discrete cosine transform (DCT) used in previous standards, is applied to the block residue, concentrating most of the block energy into the low frequency region. Quantization and entropy coding of the remaining coefficients are then applied to further reduce irrelevancy and redundancy of the video signal.

Motion estimation is among the most complex operations in H.264/AVC video coding because of the number of operations required to find the best match for each MB or block. Also on the decoder side, previous studies on H.264/AVC decoder complexity have shown that motion compensation (MC) is the most computationally complex functional block at the decoder, followed by the de-blocking filter process [8,9]. High complexity in motion compensation is mainly due to interpolation needed to decode motion vectors with half or quarter pixel accuracy. For the baseline H.264/AVC decoder, it was shown that interpolation takes around 39% of the execution time on average, and it can go up to 44% for some sequences. Figure 1 shows the breakdown of the complexity of a typical H.264/AVC decoder implementation as reported in [9]. Since this is an important topic in the context of this paper, a more detailed description of sub-pixel interpolation in H.264/AVC and the origin of its computational complexity is provided in the next subsection.

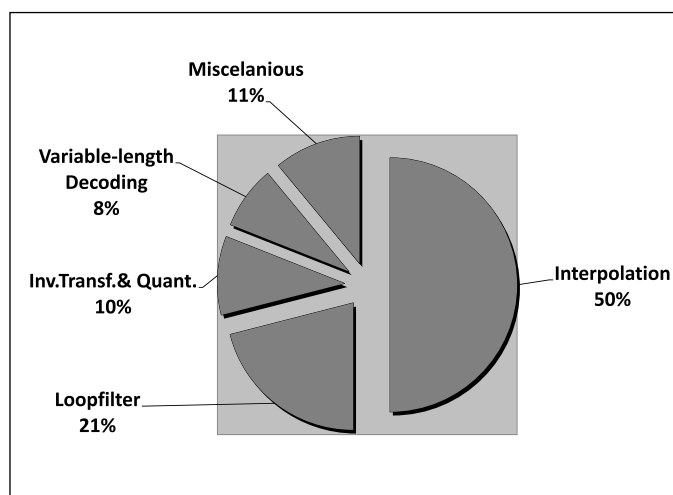


Fig. 1. Computational complexity distribution in a typical H.264/AVC decoding process [9]

2.1 Half and Quarter-Pixel Interpolation

When at decoder side are received motion vectors pointing to locations which are not integer positions on sampling grid of image, new sub-pixel values must be computed using the interpolation method in order to obtain the same prediction block that was prior calculated and used in the encoder part. Such an interpolation process is implemented through filters defined in the standard which lead to either half or quarter pixel accuracy. H.264/AVC uses up to quarter pixel precision during interpolation [10]. Figure 2 illustrates the details of sub-pixel interpolation. Shaded squares with capital letters are the integer locations, corresponding to the sampling grid, and the white squares with lowercase letters correspond to interpolated pixels with sub-pixel accuracy.

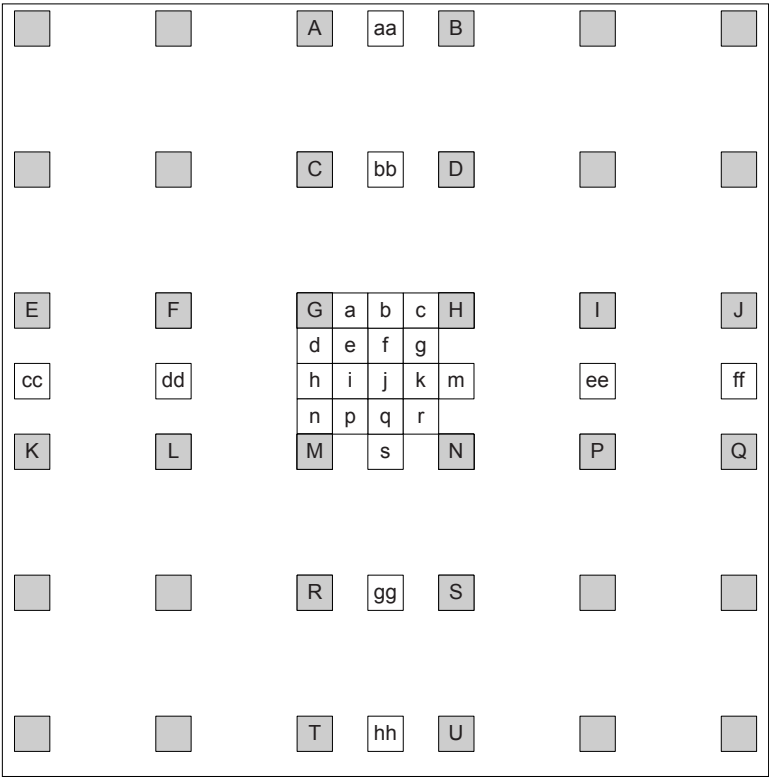


Fig. 2. Integer, half and sub-pixels locations in H.264/AVC

The values at half-pixel locations (b, h) are computed with a 6-tap FIR filter applied in both horizontal and vertical directions. All quarter-pixel values are computed by a 2-tap average filtering using both integer and half-pixels. For example, the following formulae are used to calculate sub-pixels b and p.

$$b = \frac{(E - 5F + 20H - 5I - J) + 16}{32} \quad (1)$$

$$p = \frac{h + s + 1}{2} \quad (2)$$

The total number of filtering operations required to compute each sub-pixel depends on the exact location of each one on sampling grid of image. In this work, this is closely related to the computational complexity, thus with power consumption saving in decoding devices such as portable video players.

Table 1 lists the number of taps necessary for interpolation filtering which is used as a measure of the associated complexity. From the table, it is obvious that computing different sub-pixels requires quite different computational complexity. Since this operation must be done at the decoder whenever sub-pixel accuracy motion vectors are received in the coded stream, a strong dependence between both the number and type of such sub-pixels and decoding computational complexity of P or B slices is expected.

Table 1. Interpolation filtering according to sub pixel location

Sub pixel type	Pixel location	Interpolation filtering
integer	G	No
in-line half-pels	b,h	6-tap
center half-pel	j	7*6-tap
in-line quarter-pels	a,c,d,n	6-tap + 2-tap
corner quarter-pels	p,e,g,r	2*6-tap + 2-tap
center quarter-pels	f, q, i,k	7*6-tap + 2-tap

Observing Table 1, one can establish a direct relation between the associated cost of each type of motion vector and its computational cost. The underlying idea for reducing the decoder computational complexity is to select those motion vectors that involve less interpolation operations, particularly 6-tap filtering, while keeping the video quality high. Since in general a tradeoff between only rate and distortion is used to select the best motion vector at the encoder, by introducing the decoding complexity cost in the selection process, one may optimally reduce decoding complexity without punishing too much rate-distortion performance.

Therefore the basic approach to complexity reduction is to change motion vectors from high complexity sub pixel positions into the ones with low complexity, or even to integer-pixel positions, by jointly optimize quality, bitrate, and computational complexity. This will lead to lower energy consumption in portable video players without significantly affecting the signal quality.

3 Rate-Distortion-Complexity Optimization

In video coding, the bit allocation and rate control mechanisms dynamically adjust the encoding parameters to achieve a target bit budget at the highest

possible video quality. This is done through rate-distortion (RD) Lagrangian optimization in two coding stages: motion estimation and mode decision. In the first one, motion estimation, for each block B with a block mode M, the motion vector associated with the block is selected through a rate-distortion joint cost function [11]:

$$J_{Motion}^{R,D} = D_{DFD} + \lambda_{Motion} R_{Motion} \quad (3)$$

In (3), D_{DFD} is the prediction error, computed as either the sum of absolute differences (SAD) or the sum of squared difference (SSD), R_{Motion} is the estimated bit rate to encode the corresponding motion vector, λ_{Motion} is the Lagrange multiplier to control the weight of the bit rate cost relative to the prediction error. $J_{Motion}^{R,D}$ is the rate-distortion joint cost comprising both R_{Motion} and D_{DFD} . Since in general the search space for motion vectors is very large and SAD has lighter computation cost than SSD, the former is used more often.

In this work, in order to favor those motion vectors with less interpolation complexity and penalize the ones with higher complexity, the conventional cost function presented in (3) is added a specific Lagrange term to model the complexity cost. This allows selecting the motion vectors based on a joint cost function which takes the three parameters into account, i.e., a rate-distortion-complexity joint cost function as given by (4),

$$J_{Motion}^{R,D,C} = J_{Motion}^{R,D} + \lambda_{CMotion} C_{Motion} \quad (4)$$

where C_{Motion} is the complexity cost associated with the selected motion vector and $\lambda_{CMotion}$ is the Lagrange multiplier for the complexity term. J_{Motion}^{RD} is the rate-distortion defined in (3) and $J_{Motion}^{R,D,C}$ is the rate-distortion-complexity cost function.

Considering the extreme case of $\lambda_{CMotion} = 0$ the solution for (4) is identical to that of (3), i.e. the complexity cost is not considered and motion vectors are chosen based only on the RD cost without taking into account the inherent decoding complexity of half/quarter-pixel accuracy motion. At the other extreme, where $\lambda_{CMotion} = \infty$ all motion vectors are forced to point at integer pixel locations which results in minimum decoding because no interpolation filtering is necessary. However in this case the RD performance drops, thus the best solution is to tradeoff between rate-distortion and complexity using $0 < \lambda_{CMotion} < \infty$ in (4).

3.1 Cost Model for Complexity of Half Pixel Interpolation Filtering

As pointed out before, the computational complexity is heavily influenced by the type of the motion vector (integer, half-pixel, or quarter-pixel) which defines the interpolation filters to be used in the motion compensation process. Therefore a cost model was defined to associate the decoding complexity to each candidate motion vector. Since interpolation is implemented by the predefined filters, the complexity cost was chosen to be directly related to the number of filtering operations needed in each type of interpolation. Table 2 shows the complexity costs used for each type of half/quarter-pixel location.

Table 2. Complexity cost of half/quarter-pixel interpolation

Pixel location	MV complexity cost
G	0
b,h	6
a,c,d,n	8
p,e,g,r	14
j	42
f,q,i,k	44

4 Experimental Results

The efficiency of the proposed constrained coding method was experimentally evaluated in order to assess how much quality drop is obtained for reducing the decoding complexity. Two sequences (Container and Foreman) with different types of motion were used in order to assess the influence of the video signal characteristics on the performance of the proposed method. The baseline mode of H.264/AVC was used as this is more appropriate to be delivered to mobile devices. The experimental setup was defined according to the recommended simulation conditions for coding efficiency experiments [12] and also followed the complexity evaluation methods proposed in [13]. The relevant coding parameters are listed in Table 3.

Table 3. Simulation parameters used in the experiments

Sequences Information	
Sequence name	Foreman, Mobile and Stefan
Frame size	QCIF (176 x 144)
Video format	30 fps, 300 frames
H.264 Encoder Parameters	
Prediction Structure	IPPP...
FrameSkip	1
NumberReferencesFrames	4
SymbolMode	0 (CABAC disabled)
SliceMode	0
DisableSubpelME	0
SearchRange	32
FME	0
BlockMode	All on
Complexity Simulation Parameters	
Quantization Parameters	22, 27, 32, 37
λ_{Motion}	[0 ~ 128]
RateControl	Disabled

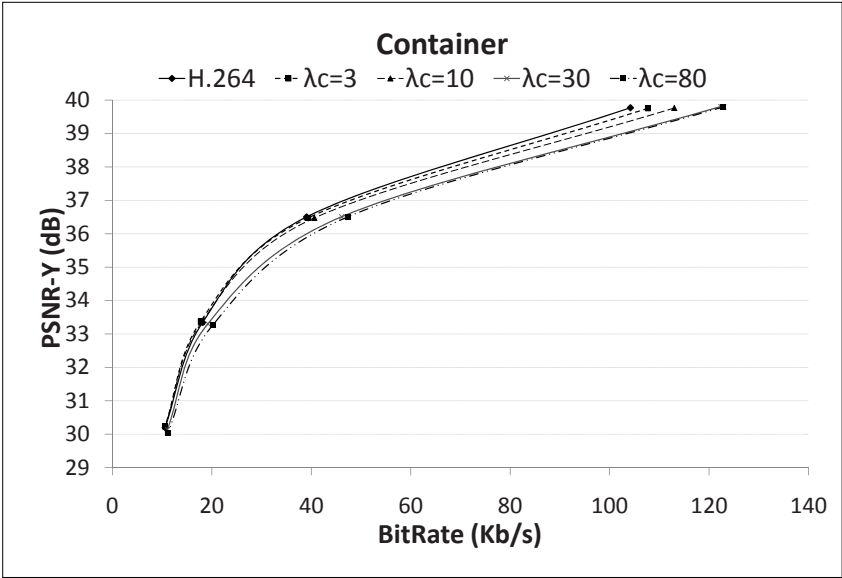


Fig. 3. Rate-PSNR performance for different values of λ_C (Container)

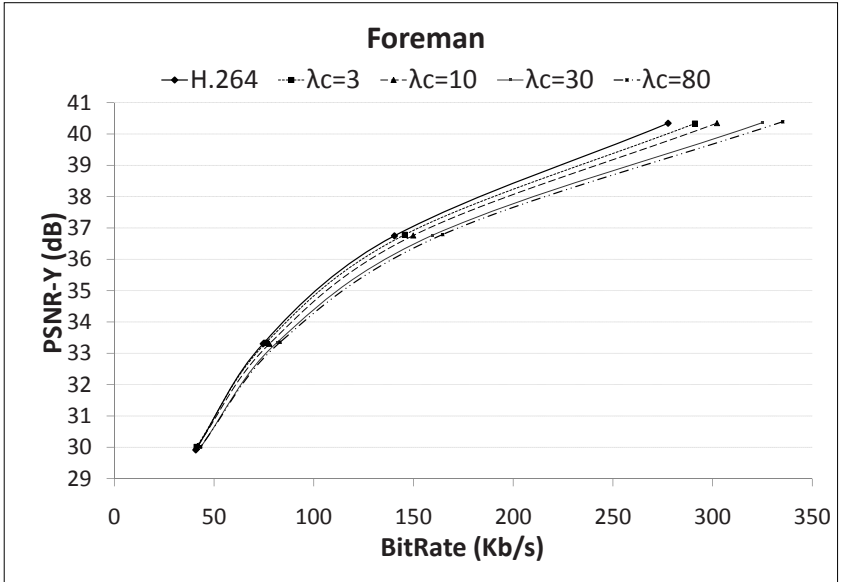


Fig. 4. Rate-PSNR performance for different values of λ_C (Foreman)

Figures 3 and 4 show the rate-PSNR performance obtained from sequences Container and Foreman for different values of C and Table 4 shows the decoding complexity measured as the processing time consumed by the motion compensation function in the H.264/AVC decoder. The proposed method is compared with H.264/AVC RD optimization, referred to as "H.264" in both figures and Table 4. As it can be seen in the figures, the proposed method produces a very small drop in PSNR for a wide range of bit rates. For higher values of C the choice for integer motion vectors increase while sub-pixel interpolation decreases because the complexity weight is higher in the Lagrangian cost function. This causes lower RD performance but significant savings in decoding computational complexity. At PSNR values close to 40 dB, i.e., at higher bit rates, the quality drop is slightly higher but the computational complexity saving is also higher. This is because at such high values of PSNR the picture quality is very high and there is almost no coefficient quantization noise to mask the small extra distortion. Nevertheless the bit rate range of interest in this type of applications is below 100 kbps, where quality drop is about 0.5 db or less and decoding complexity can be reduced up to 26%.

Table 4. Average decoding complexity

Sequence	R-D (ms)	R-D-C ($\lambda c = 3$)		R-D-C ($\lambda c = 10$)		R-D-C ($\lambda c = 30$)		R-D-C ($\lambda c = 80$)	
		(ms)	ΔC	(ms)	ΔC	(ms)	ΔC	(ms)	ΔC
Foreman	1174,6	1098,3	6,5%	1057,3	10,0%	974,3	17,1%	867,6	26,1%
Container	520,6	470,6	9,6%	457,6	12,1%	428	17,8%	440,3	15,4%

5 Conclusion

The results presented in this paper show that decoding complexity can be significantly reduced by constraining the video encoder in the choice of complex operations such as sub-pixel interpolation in the motion compensation process. By including the complexity cost in the selection procedure of the best motion vector, the rate-distortion-complexity tradeoff is optimized such that the quality loss in PSNR is perceptually negligible while the decoding complexity is significantly reduced. Therefore the proposed method is particularly useful in video services and applications which deliver H.264/AVC video to portable devices.

References

- Poellabauer, C., Schwan, K.: Power-aware video decoding using real-time event handlers. In: 5th International Workshop on Wireless Mobile Multimedia (WoW-MoM (September 2002)
- Pouwelse, J., Langendoen, K., Lagendijk, R., Sips, H.: Power-aware video decoding. In: 22nd Picture Coding Symposium, Seoul, Korea (April 2001)

3. Lin, S.S., Tseng, P.C., Lin, C.P., Chen, L.G.: Multi-mode content-aware motion estimation algorithm for power-aware video coding systems. In: IEEE Workshop on Signal Processing Systems, SIPS 2004, October 13-15, pp. 239-244 (2004)
4. Lu, M.T., Yao, J.J., Chen, H.H.: A complexity-aware video adaptation mechanism for live streaming systems. *EURASIP J. Appl. Signal Process* 2007(1), 215-215 (2007)
5. Adams, J.: Adaptive buffer power save mechanism for mobile multimedia streaming. Master's thesis, School of Electronic Engineering, Faculty of Computing and Engineering Dublin City University (July 2007)
6. Wang, Y., Chang, S.F.: Complexity adaptive h.264 encoding for light weight streams. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France (2006)
7. Kalva, H., Furht, B.: Complexity estimation of the h.264 coded video bitstreams. *Comput. J.* 48(5), 504-513 (2005)
8. Horowitz, M., Joch, A., Kossentini, F., Hallapuro, A.: H.264/avc baseline profile decoder complexity analysis. *IEEE Trans. Circuits Syst. Video Techn.* 13(7), 704-716 (2003)
9. Lappalainen, V., Hallapuro, A., Hämäläinen, T.D.: Complexity of optimized h.261 video decoder implementation. *IEEE Trans. Circuits Syst. Video Techn.* 13(7), 717-725 (2003)
10. Wiegand, T., Sullivan, G.J., Bjntegaard, G., Luthra, A.: Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13(7), 560-576 (2003)
11. Sullivan, G., Wiegand, T.: Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine* 15(6), 74-90 (1998)
12. Tan, T., Sullivan, G., Wedi, T.: Recommended simulation common conditions for coding efficiency experiments. Revision 1. ITU-T Video Coding Experts Group (ITU-T SG16 Q.16), Document VCEG-AE010, 31st Meeting: Morocco (January 2007)
13. Horowitz, M.: Towards useful complexity evaluation methods. Document VCEG-AG19, 33rd meeting: Shenzhen, China (October 2007)