FAST MODE DECISION WITH EARLY TERMINATION FOR H.264/AVC VIDEO CODING

Y.V. Ivanov, C. J. Bleakley

School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland
e-mail: {yury.ivanov, chris.bleakley}@ucd.ie

Keywords: Mode decision, H.264, advanced video coding.

Abstract

In this paper a fast mode decision algorithm based on an early termination procedure is proposed for H.264/AVC video encoding. Unlike previous methods, the termination decision is based on the Rate Distortion cost function. A statistical analysis of the spatio-temporal characteristics of the Rate Distortion cost function and of the probability of mode transition is given for test sequences. Experimental results show that the new algorithm provides a 38% reduction in total computational complexity with a negligible increase in the bit rate and negligible reduction in visual quality when compared to conventional encoding.

1 Introduction

There is currently considerable commercial interest in providing video playback and recording features on mobile devices such as cell phones, handheld game consoles and music players. The H.264 standard [1] developed by the Joint Video Team (JVT) provides better coding efficiency than MPEG-2 and H.263 at lower bit rates. However, this coding efficiency is achieved at the cost of a tenfold increase in computational complexity [2, 3]. In mobile devices, this leads to a significant reduction in battery lifetime.

H.264 provides seven Variable Block Size (VBS) modes for Motion Estimation (ME). The multi-block segmentation procedure splits large macroblocks (MB) into smaller partitions each having a separate motion vector [2]. Mode Decision (MD) processing is a significant portion of encoder complexity. Processing time increases linearly with each additional VBS mode used for segmentation [3]. Full VBS search provides superior coding results but complexity is increased by 2.5–3 times relative to use of only the largest (16x16) block size [3].

It has been shown that the conventional Mode Decision technique can be significantly improved in terms of computational complexity by using a so-called "early termination" procedure. If some VBS modes can be dropped from examination earlier by predicting that they will never be chosen in the final decision then computational complexity will be reduced.

The key to successful early termination algorithms is in using an efficient metric for the termination decision. Incorrect termination decisions cause poor MB segmentation which leads to bit rate increases and visual quality degradation.

This paper presents a novel fast MD algorithm that utilizes the Lagrangian Rate-Distortion (RD) cost function J as a metric for the early termination procedure. Experimental results show that using J outperforms previously proposed metrics such as Sum of Absolute Differences (SAD) or motion intensity. Compared to the reference encoder computational complexity is significantly reduced, bit rate is slightly reduced and visual quality loss is negligible.

The paper is organized as follows. Section 2 gives a brief overview of MD in H.264. Section 3 reviews related work in the field. Section 4 provides a statistical analysis of the RD cost function and of mode transitions. The new fast MD algorithm is described in Section 5 and experimental results are presented in Section 6. Section 7 concludes the paper.

2 Overview of mode decision in H.264

H.264 Mode Decision minimizes the Lagrangian Rate-Distortion cost function [4]:

$$\min\left\{J(s,c,MODE\,|\,QP,\lambda_{MODE})\right\} \tag{1}$$
$$J(s,c,MODE\,|\,QP,\lambda_{MODE}) = SSD(s,c,MODE\,|\,QP) + \lambda_{MODE}\cdot R(s,c,MODE\,|\,QP)$$

where QP is the macroblock quantization parameter and λ_{MODE} is the Lagrange multiplier. $MODE$ is the mode chosen from the set of potential modes:

$$MODE \in \{INTRA4\times4,\ INTRA16\times16,\ SKIP,\ 16\times16,\ 16\times8,\ 8\times16,\ 8\times8,\ 8\times4,\ 4\times8,\ 4\times4\} \tag{2}$$

R($s,c,MODE|QP$) is the transmitted bit rate associated with $MODE$ and QP. SSD is the Sum of the Squared Differences between the original block s and its reconstruction c:

$$SSD(s,c,MODE\,|\,QP) = \sum_{x=1,y=1}^{16,16}\left(s\left[x,y\right] - c\left[x,y,MODE\,|\,QP\right]\right)^2 \tag{3}$$

where $c[x,y,MODE|QP]$ and $s[x,y]$ represent the reconstructed and original pixel values. Finally, the Lagrangian multiplier λ_{MODE} is given as:

$$\lambda_{MODE} = 0.85 \cdot 2^{(QP-12)/3} \qquad (4)$$

The MD algorithm [5] implemented in the reference JM encoder [6] simply calculates *J* for all segmentation options. The algorithm selects the VBS mode giving minimum *J*.

3 Related work

Several fast mode decision algorithms for H.264 have been proposed in the literature [8-13]. All are variants of the early termination approach utilizing various metrics for MB mode prediction.

The algorithm proposed in [8] utilizes a special block matching order combined with SAD pre-calculation for reducing ME complexity and for skipping spatial predictive coding. The method in [9] is based on the correlation of motion vectors across the various MB partitions, but is much less effective.

The authors of [10] base their algorithm on MB mode correlation and classify MBs according to their motion-energy ratio, which is used for MB mode prediction and partial RDO decision. It operates in one hybrid RDO mode in contrast to the most other methods that can operate in two modes (i.e. RDO on/off).

The MD algorithm in [11] uses a threshold-based termination decision with early SKIP decision. The authors of [12] propose calculation of the RD cost for 8x8 and 16x16 macroblocks and use a combined metric based on these values.

The algorithm described in [13] uses VBS prediction from the surrounding MBs. The method suffers from the disadvantage that block modes are predicted from frame border MBs.

Comparison of the complexity figures quoted in the literature for these algorithms is problematic. Execution time is given as a percentage relative to various versions of the JM reference encoder. Encoding configurations vary and are often not well specified in all papers, e.g. RDO on or off, search range 8-32, CALVC or CABAC. Two algorithms may have equal complexity if run in the same experimental environment but the reported complexity figures may vary greatly.

The most effective algorithms are based on the SAD metric. It is worth noting that the RD cost function metric *J* is based on SSD which has similar properties but higher perceptual correlation [14]. Use of *J* has the advantage that is calculated a part of the MD process and does not require an additional processing step.

4 Statistical analysis

In the first series of experiments we investigated if previously calculated values of the Rate-Distortion cost function can be used as a prediction metric for the early termination procedure. Pearson correlation coefficients were calculated between *J* values obtained during full search for blocks and their neighbors in the same and the previous frame. In the experiments the optimal *J*, i.e. the best *J* for each macroblock was used. The correlation coefficient for neighboring block *j* was calculated according to the following formula:

$$r_j = \frac{\sum_{i=1}^{N} \left(X_i - M_X\right)\left(Y_i - M_Y\right)}{(N-1) \cdot S_X \cdot S_Y} \qquad (5)$$

where *N* is the number of 16x16 blocks in the video sequence, X_i is the *J* value for the current block and Y_i is *J* value for the neighboring block. For each value of *j*, the position of the neighboring block is fixed relative to the current block (e.g. previous frame, one MB up). M_X, S_X, M_Y, and S_Y are the mean and standard deviation of X_i and Y_i respectively.

Video sequences with different content and motion intensity were encoded using the standard JM encoder as described in Section 6. Results for two typical video sequences are provided in Figure 1.

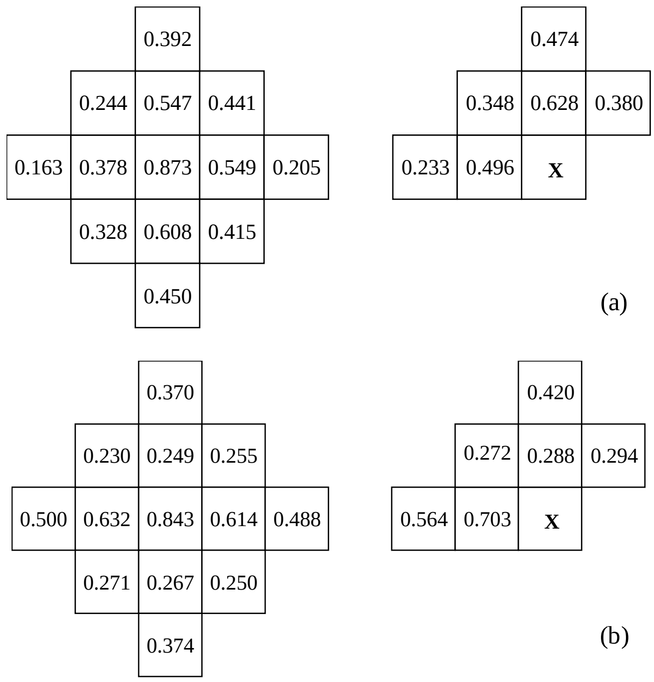

Figure 1: Correlation coefficients r_j for the current frame (right) and the previous frame (left) for: (a) Foreman, QCIF and (b) Coastguard, CIF. X indicates the position of the current block.

It can be clearly seen, that the RD cost function *J* is highly correlated between neighboring blocks. As an example, correlation coefficients between for motion vectors were reported in [7] to be around 0.4 for blocks in the same frame and around 0.15 for blocks in the previous frame. Correlation values between *J* are much higher. Therefore *J* is suitable as a metric for mode prediction.

In the next experiments, an analysis was performed of the probability of block mode transition from one frame to the next with the same encoding configuration as in previous experiments. Results are shown in Table 1.

from\to		*Skip*	*Inter*				*Intra*	
			16x16	*16x8*	*8x16*	*8x8*	*4x4*	*16x16*
Skip		85.9	10.2	0.7	0.8	1.4	0.0	0.7
Inter	16x16	40.5	40.8	3.3	3.5	10.6	0.2	0.9
	16x8	27.1	33.4	10.9	6.0	20.5	0.7	1.1
	8x16	27.4	31.0	5.2	12.2	22.3	0.8	0.8
	8x8	9.6	20.4	3.7	4.4	59.7	1.6	0.3
Intra	4x4	25.1	12.9	2.6	3.6	25.4	24.4	5.6
	16x16	67.2	17.2	1.5	1.0	1.7	1.7	9.5

Table 1: Transition probabilities for different MB modes, %.

The lowest probabilities are for transitions from Inter modes to Intra, thus, if found, these transitions can be disabled.

We base our algorithm on the results of the given analysis.

5 Fast mode decision algorithm

In contrast to existing fast mode decision methods, the new algorithm exploits the spatio-temporal statistics of the Lagrangian RD cost function *J*. The algorithm utilizes *J* values from the previous frame in order to omit unnecessarily MD computations. In addition, the MD for all MBs is stored in order to identify modes that can be excluded from the search due to low transition probability. The algorithm is summarized in the following pseudo-code:

```
prev_mode[all MBs] = intra_4x4;
mean_J = 0;
mean_J_intra = 0;
for (all frames)
if (first I-frame)
{ calculate mean_J_intra for all MBs;
  perform normal MD and MB encoding;
}
else
if (first P-frame)
{ calculate mean_J for all MBs;
  perform normal MD and MB encoding;
}
else // perform early termination MD
 for (all MBs)
    end_search = false;
    J[all modes] = maximum;
    for (all modes or until end_search){
      if (mode transition is enabled) {
        encode MB with current mode;
        calculate J[mode];
        if (inter & J[mode] <= mean_J)
          end_search = true;
        if (intra & J[mode] <= mean_Ji)
          end_search = true;
        if (end_search) {
          best_J = J[mode];
          best_mode = mode;
        }
      }
      if (not end_search) {
        best_J = minimum J[mode];
        best_mode = mode with best_J;
      }
      transmit MB using best_mode;
      prev_J[MB] = best_J;
      prev_mode[MB] = best_mode;
    }
  }
  mean_J = mean of prev_J for all MBs;
  mean_J_intra = mean of prev_J for
                 intra MBs only;
}
```

Mode search is in the order Inter16x16, Inter16x8, Inter8x16, Inter8x8, Intra4x4, Intra6x16, Skip. The following mode transitions are disabled: Skip, Inter16x16, Inter16x8, Inter8x16, or Intra16x16 to Intra4x4 and Inter8x8 to Intra16x16 (see Table 1).

Despite the high local correlation found for *J* (see Figure 1), the algorithm uses mean *J* across the whole frame in order to avoid localizing of high and low quality areas (or ‘convergence problem’) and produce more even results.

Note that first early termination part of the algorithm starts to operate only after mean *J* for I- and P- macroblocks had been calculated. Thus, for the selected “IPPP” GOP structure (see experiments below) first two frames (I- and P-) are processed with standard mode decision algorithm.

6 Experimental results

In the experiments, the performance of the proposed method was compared with that of the reference JM 9.5 encoder developed by JVT. QCIF and CIF video sequences of different motion content were used. Each consisted of 300 frames encoded at 30 fps with GOP structure, when the initial frame in the video sequence was encoded as I-frame and the rest of the frames as P-frames. Reference encoding used all seven VBS modes, search range size of 8, CABAC entropy coder and RDO off. Experiments were performed on a 3GHz Pentium 4 PC for *QP* settings ranging from 26 to 32. The results are provided in Table 2.

Bit rate increase $\Delta Bits$, quality degradation $\Delta PSNR$ and execution time change Δt are calculated according to equations (6)–(8), where the minus sign (–) means improvement for the new method. Note, that total encoding time is used in equation (8), not MD time.

$$\Delta Bits = \frac{Bits_{method} - Bits_{JM}}{Bits_{JM}} \cdot 100\% \quad (6)$$

$$\Delta PSNR = PSNR_{JM} - PSNR_{method} \quad (7)$$

$$\Delta t = \frac{t_{method} - t_{JM}}{t_{JM}} \cdot 100\% \quad (8)$$

From Table 2, the total encoding time for the new method shows a reduction of almost 40% on average (38.31%). High motion sequence "Carphone" shows about 35% of average complexity reduction, while for the low motion sequences (e.g. Akiyo, Hall) complexity reduced about 40–43%.

Average bit rate increase is almost zero (0.05%), for some sequences it is even lower than for the original JM (e.g. Container, Hall, Akiyo). Quality drop is insignificant (only 0.05dB). Visual examination of decoded video sequences did not reveal any anomalies or blocking artefacts. In fact, differences between images produced by the original JM encoder and the encoder using the new method can hardly be found.

Video sequence	*QP*	*ΔBits, %*	*ΔPSNR, dB*	*Δt Total, %*
Carphone, QCIF	26	1.0	0.11	-36.26
	28	0.54	0.14	-34.48
	30	0.51	0.10	-33.91
	32	-0.05	0.12	-34.11
Container, QCIF	26	0.12	0.07	-33.92
	28	0.20	0.10	-36.82
	30	0	0.07	-38.24
	32	-1.46	0.10	-38.22
Table, QCIF	26	2.25	0.12	-39.88
	28	0.79	0.08	-38.44
	30	1.04	0.04	-38.94
	32	0.68	0.08	-38.41
Paris, CIF	26	0.41	0.05	-38.85
	28	0.40	0.06	-37.79
	30	0.38	0.05	-37.54
	32	0.22	0.06	-35.64
Hall, CIF	26	0.22	0.04	-43.92
	28	-0.06	0.05	-42.76
	30	-0.57	0.01	-41.54
	32	-0.46	0	-39.95
Akiyo, CIF	26	-0.74	0	-40.48
	28	-0.38	0	-40.90
	30	-0.71	-0.20	-39.76
	32	-2.96	0	-38.68
Mean		***0.05%***	***0.05dB***	***-38.31%***

Table 2: Performance comparison of the proposed fast MD versus reference JM.

Rate-Distortion performance of the new algorithm and reference JM can be also compared from the Figure 2. It can be seen, that the reduced complexity algorithm has negligible effect on rate-distortion performance.

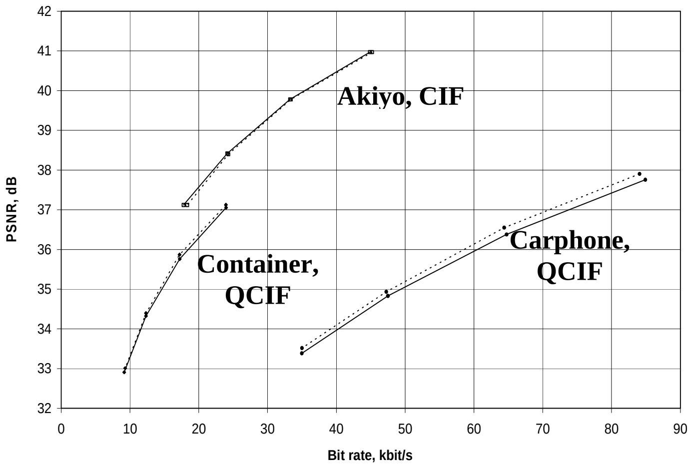

Figure 2: Rate-Distortion performance of the reference JM (shown in dashed lines) and fast MD (shown in solid lines).

PSNR and bit rate change for the tested video sequences was also calculated using the Bjontegaard delta method [15]. The results are given in the Table 3 below. The minus sign (–) indicates that the new algorithm outperforms the standard encoder.

Video sequence	*Bjontegaard ΔPSNR, dB*	*Bjontegaard ΔBR, %*
Carphone, QCIF	0.42	8.40
Container, QCIF	-0.02	-0.43
Table, QCIF	0.02	0.40
Paris, CIF	0.07	1.58
Hall, CIF	-0.15	-3.1
Akiyo, CIF	-0.19	-3.83

Table 3: PSNR and bit rate change calculated using Bjontegaard delta method [15].

The complexity of the fast MD method is roughly equivalent to performing MD using on average less than 2 VBS modes instead of 7. The results show that the method outperforms previously reported fast MD algorithms in terms of computational complexity, bit rate and visual quality.

7 Conclusions

The proposed fast mode decision algorithm, based on early termination using the RD cost function, can significantly reduce the computational complexity of H.264 for Intra- and Inter- MB encoding, while visual quality and bit rate is maintained.

Correlation between RD cost function values for neighbouring blocks and mode transition probabilities have been investigated in a series of experiments.

Future work includes development of algorithms to predict SKIP decisions and study of other RD-based mode predictors.

Acknowledgements

This research project is sponsored by University College Dublin, Ireland.

References

[1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H264|ISO/IEC 14496-10 AVC) ", document JVT-G050d35.doc, 7th Meeting: Pattaya, Thailand, (2003).

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. "Overview of the H.264/AVC Video Coding Standard", *IEEE Trans. Circuits Syst. Video Technol.*, **vol. 13**, no. 7, pp. 560–576, (2003).

[3] Implementation Studies Group of ISO/IEC. "Main Results of the AVC Complexity Analysis", *ISO/IEC JTC1/SC29/WG11*, Klagenfurt, (2002).

[4] H. Everett III. "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", *Operations Research*, **vol. 11**, pp. 399–417, (1963).

[5] ITU-T Q.15/SG16. "H.26L Test Model Long Term Number5 (TML-5) draft0", document Q15-K-59, (2000).

[6] JVT reference software JM 9.5, on the Web: http://iphome.hhi.de/suehring/tml.

[7] F. Kossentini, Y.W. Lee, M.J.T. Smith and R.K. Ward. "Predictive RD Optimized Motion Estimation for Very Low Bit-Rate Video Coding", *IEEE Journal on Selected Areas in Comm.*, **vol.15**, no. 9, pp. 1752–1763, (1997).

[8] K. Han and Y. Lee. "Fast Macroblock Mode Decision in H.264", *Proc. IEEE Region 10 Conf. (TENCON'04)*, **vol.1**, pp. 347 – 350, (2004).

[9] T. Y. Kuo and C.H. Chan "Fast Macroblock Partition Prediction for H.264/AVC", *Proc. IEEE Int. Conf. Multimedia and Expo (ICME'04)*, **vol. 1**, pp. 675–678, (2004).

[10] Z. Hong, W. Cheng-Ke, W. Yang-Li and F. Yong. "Fast Mode Decision for H.264/AVC based on Macroblock Correlation", *Proc. IEEE 19th Int. Conf. Advanced Information Networking and Applications (AINA'05)*, **vol. 1**, pp. 775–780, (2005).

[11] P. Yin, H.Y. C. Tourapis, A. M. Tourapis and J. Boyce. "Fast Mode Decision and Motion Estimation for JVT/H.264", *Proc. IEEE Int. Conf. Image Processing (ICIP'03)*, **vol.3**, pp. 853–856, (2003).

[12] J. Lee and B. Jeon. "Fast Mode Decision for H.264 with Variable Motion Block Sizes", *Proc. Int. Symp. Computer and Information Sciences (ISCIS'03)*, pp. 723–730, (2003).

[13] G. L. Li, M. J. Chen, H. J. Li and C. T. Hsu. "Efficient Motion Search and Mode Prediction Algorithms for Motion Estimation in H.264/AVC", *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'05)*, **vol. 6**, pp. 5481 – 5484, (2005).

[14] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architecture for MPEG-4 Motion Estimation*, Kluwer Academic Publishers, Boston, (1999).

[15] G.Bjontegaard, "Calculation of average PSNR differences between RD curves", ITU-T VCEG Meeting, document VCEG-M33, Austin, TX, (2001).