Temporal-Spatial Correlation Based Mode Decision Algorithm for H.264/AVC Encoder

Bin Zhan, Baochun Hou and Reza Sotudeh
School of Electronic, Communication and Electrical Engineering,
University of Hertfordshire, Hatfield, United Kingdom
E-mail: {b.zhan, b.hou, r.sotudeh}@herts.ac.uk

Abstract

The latest emerging video compression standard, H.264/AVC, achieves better compression performance than previous standards. Variable block size based motion estimation (ME) with optimized mode decision (MD) is one of the most important improvements leading to high quality and low bitrate encoding results. However, such improvement is at the cost of remarkably complex calculation, which is a major bottleneck for some applications, especially in wireless communications. A new MD algorithm is presented in this paper. Taking into account the correlation between macroblocks, the computational complexity is reduced significantly by effectively skipping some unnecessary modes. Experimental results show that the encoding time can be reduced up to 55.26% with negligible loss of encoding efficiency.

1. Introduction

The new video compression standard, H.264 [1], outperforms the former standards in coding efficiency by employing several new techniques [2, 3]. Flexible block sizes are introduced as one of the impressive features. Each macroblock (16×16) can be further partitioned into smaller blocks: {16×8, 8×16, 8×8, 8×4, 4×8 and 4×4}. Variable block size helps the encoder to more accurately and effectively predict the contents in video frames, especially, there are high detailed parts contained. Rate distortion optimization (RDO) based mode decision (MD) balances residuals and motion vector (MV) to choose the most best mode, which minimises the Lagrangian cost. The Lagrangian function is given in (1):

$$J(s,c,MODE/QP,\lambda_{MODE}) = SSD + \lambda_{MODE} \cdot R \quad (1)$$

where *MODE* is one of the candidate modes, *QP* is the quantization parameter. The Lagrangian multiplier for MD, λ_{MODE}, is defined in (2):

$$\lambda_{MODE} = \begin{cases} 0.85 \times 2^{QP/3} & \text{P-frame} \\ \max(2, \min(4, \frac{QP}{6})) \times \lambda_{MODE_P} & \text{B-frame} \end{cases} \quad (2)$$

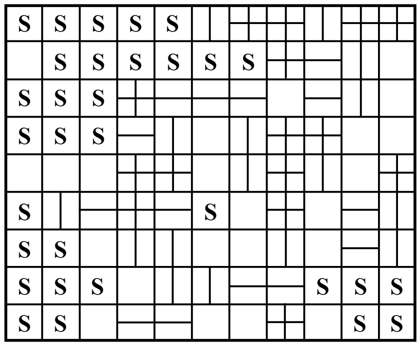
The sum of squared differences (*SSD*) gives the distortions between current macroblock (s) and reference macroblock (c), and its calculation is given in (3). *R(s, c, MODE/QP)* denotes the rate for current mode.

$$\begin{aligned} SSD(s,c,MODE|QP) &= \sum_{x=1,y=1}^{16,16}(s_Y[x,y]-c_Y[x,y,MODE|QP])^2 \\ &+ \sum_{x=1,y=1}^{8,8}(s_U[x,y]-c_U[x,y,MODE|QP])^2 \\ &+ \sum_{x=1,y=1}^{8,8}(s_V[x,y]-c_V[x,y,MODE|QP])^2 \end{aligned} \quad (3)$$

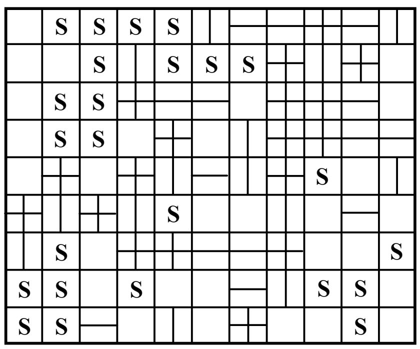
Obviously, complicated calculations of *J* consume huge amount of time in the encoding process. For each candidate mode, the calculation of Lagrangian cost has to be repeated. Mobile video communication is one of the major application areas of H.264, which is difficult to implement due to the computational complexity for real-time and low power consumption requirements. Researchers [4-9] indicated that RDO enabled MD consumes the high percentage of encoding time, and some fast MD algorithms are also introduced to eliminate the unnecessary mode calculation in exhaustive MD. The temporal-spatial correlation in MD is investigated and a new efficient algorithm is proposed in this paper. By exploiting the correlation between macroblocks, 36.85% encoding time can be saved on average, and the degradation of encoding efficiency in terms of video quality and compression ratio is negligible, especially for those video sequences with low-speed motions.

This paper is organized as follows. The next section introduces the main principles of proposed MD

0-7695-3110-5/08 $25.00 © 2008 IEEE
DOI 10.1109/DELTA.2008.99



frame$_{(t-1)}$



frame$_{(t)}$

*where **S** indicates the SKIP mode*

Figure 1: Macroblock mode decision of two successive QCIF frames (Foreman).

algorithm, and the detailed algorithm is given in section 3. The section 4 shows the simulation results. The conclusion of this paper is drawn thereafter.

2. Principles

2.1. Temporal-spatial correlation

Normally, video sequence is continued in both temporal domain and spatial domain, and this characteristic can be utilized to remove the redundant information from the video sequence in order to achieve the significant compression. On one side, the objects in successive frames give smoother movement due to the higher frequency of temporal sampling. By using the previously encoded frames to predict the motion of objects in current frame, the temporal redundancy can be removed, and that is called inter-prediction. On the other side, the high-density sampling points in each frame leads to the spatial correlation between the adjacent grids, and that is the motivation behind intra-prediction. Our investigation

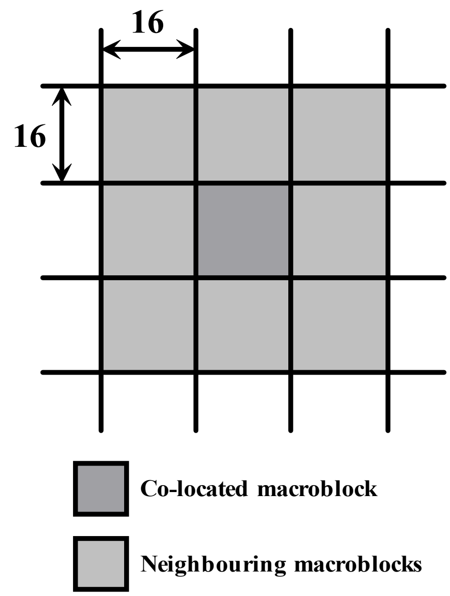


Figure 2: Neighbouring macroblocks**.**

shows that the temporal-spatial correlation also exists in MD. An example is given in figure 1, which illustrates the best mode for each macroblock in two successively encoded frames, where frame$_{(t-1)}$ is previously encoded and used as the reference for current frame$_{(t)}$.

From figure 1, it is clearly seen that most of the macroblocks in frame$_{(t)}$ possess same modes as their co-located or neighbouring macroblocks in frame$_{(t-1)}$. The co-located and neighbouring macroblocks are defined in figure 2. For the exhaustive MD algorithm seven modes will be used to form the candidate mode list $L_{(x, y)}$ to encode current macroblock (x, y). There exist some redundant modes in $L_{(x, y)}$ according to the above example. If some low-probability modes can be removed from $L_{(x, y)}$, the computational complexity of MD could be simplified. Simulations indicate that for some video sequences with low-speed motions, a lot of modes could be excluded and the degradation of encoding efficiency in terms of PSNR and bitrate is negligible. Based on the above analysis, a basic method is proposed to dynamically adjust the number of modes in $L_{(x, y)}$. For most of macroblocks in the current frame, the best modes of previously encoded macroblocks in the reference frame(s) will be used to generate the candidate mode list $L_{(x, y)}$. Only the modes in $L_{(x, y)}$ are used for MD, therefore the encoding time could be reduced significantly. For those macroblocks on the edge of the frame, there are not sufficient reference macroblocks to generate $L_{(x, y)}$. In order to avoid the degradation of encoding performance, the exhaustive MD algorithm is deployed for the macroblocks on the edge.

2.2. References for MD

Since the video sequence is temporally continued, the most previously encoded frame will be used as the reference to generate the candidate modes list for macroblocks in the current frame. Furthermore, in order to minimise the continuous downgrade of encoding efficiency incurred by the improper MD strategy adopted in the previously encoded frames, the first of every 30 frames (the number depends on the frame rate) will be used as extra reference for the proposed MD algorithm.

2.3. Motions

In most cases, the values of MVs provide information on the speed and the direction of movement of objects. Smaller values are obtained for static background and smoothly moving objects, and larger values indicate the speedy motion. Taking into account the real life videos, the high-speed movement of objects happens from time to time. In these cases, the method introduced in section 2.1 could exclude the best mode from the dynamic list, and the encoding performance will be degraded if there is no further adjustment scheme. In the proposed algorithm, MV is exploited to form the criterion to adjust the current dynamic candidate mode list. We defined the displacement of MVs as the threshold to judge the status of movement:

if $MV_x >= TH$ *or* $MV_y >= TH$ *, then high-speed motion*

where x is for horizontal direction and y is for vertical direction. For simplicity, the best MVs for inter 16×16 mode are checked only. A flag will be set for the macroblock if MV exceeds the threshold. While encoding a macroblock, the fast MD strategy will be available if no flag is set in the co-located macroblock from the reference frame; otherwise, all modes will be enabled for the current macroblock.

3. The proposed algorithm

The detailed algorithm is outlined as following:

Step A: Set the frame counter *Count* = 0. Initialize the reference tables T_0 and T_1 to record the selected modes, where T_0 stores the selected modes of when *Count* = 1, and T_1 stores the selected modes of the most previously encoded frame.

Step B: Encode the I-frame, *Count* = *Count* + 1.

Step C: If (*Count* = = 1)
{
Encode the current macroblock $M_C(x, y)$ using all modes as specified in H.264, and record the selected mode in T_0;
If ($MV_{16\times16}$ >= 5 in any direction)
{
Flag(x, y) = High;
}
Else
{
Flag(x, y) = Low;
}
Repeat until the last macroblock of the current frame has been encoded.
Skip to Step F;
}
Else, go to Step D.

Step D: If ($M_C(x, y)$ is on the edge of frame)
{
Encode $M_C(x, y)$ by using all modes;
}
Else if (*Flag(x, y)* = = High)
{
Encode $M_C(x, y)$ by using all modes;
}
Else
{
Check the co-located and neighbouring positions in T_0 and T_1, then create a dynamic candidate mode list $L_{(x, y)}$. Use $L_{(x, y)}$ to encode $M_C(x, y)$;
}
Update *Flag(x, y)*.

Step E: If (the current macroblock is not the last one of the frame), go back to Step D;
Otherwise, go to Step F.

Step F: If (the current frame is the last one of the video sequence)
{
The encoding process is terminated;
}
Else if (*Count* = = 30)
{
Count = 1;
}
Else
{
Count = *Count* + 1;
}
Go back to Step C.

Table 2: Simulation results.

		B1	B2	C1	C2	C3	C4	F	H	M	S
QP 28	ΔT	-53.45	-53.42	-23.29	-43.09	-45.82	-20.34	-19.40	-31.24	-28.68	-33.04
	ΔP	-0.02	-0.02	0	-0.03	-0.04	-0.02	-0.02	-0.03	0	-0.07
	ΔB	-0.27	-0.78	-0.09	0.65	1.30	0.24	0.36	0.79	0.07	0.44
QP 32	ΔT	-53.44	-54.60	-21.98	-44.88	-43.93	-22.92	-19.19	-36.25	-26.45	-33.61
	ΔP	0	-0.02	-0.04	-0.09	-0.05	-0.05	-0.03	-0.05	-0.02	-0.01
	ΔB	0	0.46	-0.05	2.07	1.68	0.23	0.29	0.55	-0.02	0.72
QP 36	ΔT	-53.01	-55.26	-26.08	-43.63	-42.39	-27.55	-21.61	-40.51	-24.89	-34.54
	ΔP	0d	-0.01	-0.02	-0.13	-0.11	-0.06	-0.05	-0.03	-0.01	-0.06
	ΔB	0	0.99	0.55	1.14	1.38	0.72	0.47	0.71	0.57	0.94
QP 40	ΔT	-53.36	-53.60	-28.44	-43.72	-42.02	-34.57	-28.42	-45.73	-25.27	-40.22
	ΔP	0	-0.02	-0.02	-0.16	0.02	-0.09	-0.05	-0.11	-0.04	-0.04
	ΔB	0	0.21	-0.37	1.19	1.51	0.92	-0.28	0.62	0.65	0.82

ΔT: ΔTime (%), ΔP: ΔPSNR (dB), ΔB: ΔBitrate (%)

4. Simulation results

The assessment of the proposed MD algorithm was conducted using the reference software JM10.1 [10], and the proposed algorithm was also integrated into it. Ten commonly used test video sequences are selected in the evaluations: {bridge-far(B1), bridge-close(B2), coastguard(C1), Claire(C2), container(C3), carphone (C4), foreman(F), highway(H), mobile(M), and silent(S)}. The simulation environments are summarized in table 1:

Table 1: Simulation environment.

CPU	Intel Pentium IV 3.0 GHz
Memory	1 Gb memory
System	Windows XP Professional
Profile	Baseline
Search range	[-16, 16]
GOP type	IPPP
ME scheme	Full search
Reference	One frame
Frame format	QCIF
Length of Sequence	300 frames

The encoding efficiency of the proposed algorithm is evaluated according to three parameters: the encoding time saving rate Δ*Time*, the variation of video quality Δ*PSNR* and the undulation rate of bitrate Δ*Bitrate*, which are defined in (4), (5) and (6):

$$\Delta Time = \frac{Time_{proposed} - Time_{JM10.1}}{Time_{JM10.1}} \times 100\% \quad (4)$$

$$\Delta PSNR = PSNR_{proposed} - PSNR_{JM10.1} \quad (5)$$

$$\Delta Bitrate = \frac{Bitrate_{proposed} - Bitrate_{JM10.1}}{Bitrate_{JM10.1}} \times 100\% \quad (6)$$

Table 2 shows the simulation results based on the different quantization parameters (QP). From the results, the proposed MD algorithm averagely saves 36.85% encoding time. For most of test sequences, the decrease of video quality is less than 0.1 dB, and the increase of bitrate is within 1%. Figure 3, figure 4 and figure 5 illustrate the PSNR values with the increase of bitrate for three sequences, which represent slow, medium and fast movement:

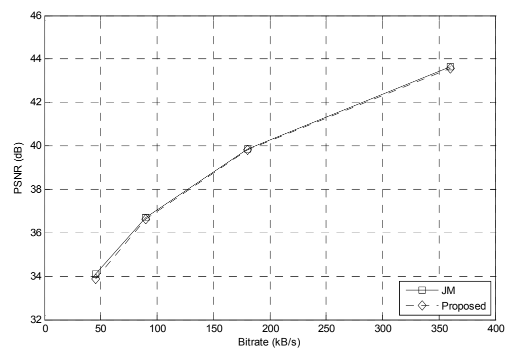


Figure 3: Rate-distortion curve for Container**.**

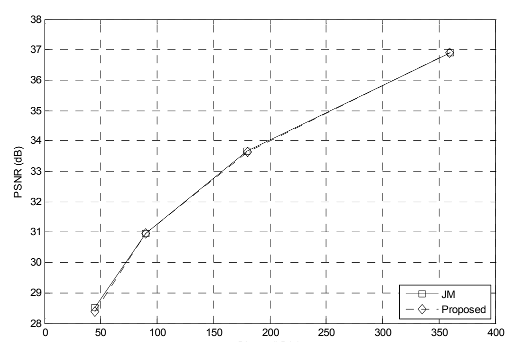


Figure 4: Rate-distortion curve for Foreman**.**

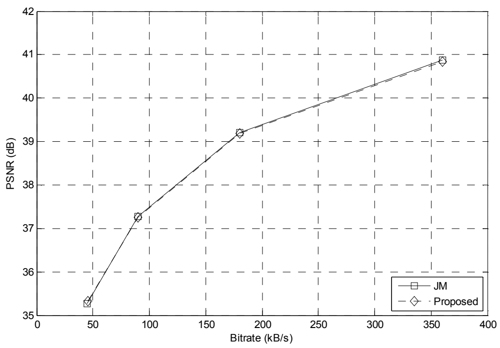


Figure 5: Rate-distortion curve for Highway**.**

In [4], Grecos gave the simulation results under the same configuration of JM software. The comparison between [4] and the proposed algorithm is also given in table 3.

Table 3: Comparison with Grecos's proposed algorithm (QP = 28).

Sequence	MD	ΔT (%)	ΔP (dB)	ΔB (%)
silent	[4]	-37.68	-0.08	-0.62
	proposed	-33.04	-0.07	0.44
container	[4]	-45.04	-0.11	-0.11
	proposed	-45.82	-0.04	1.30
bridge-far	[4]	-72.82	-0.05	0
	proposed	-53.45	-0.02	-0.27
foreman	[4]	-12.50	0	0.67
	proposed	-19.40	-0.02	0.36
carphone	[4]	-13.21	-0.11	-0.13
	proposed	-20.34	-0.02	0.24
coastguard	[4]	-11.70	-0.01	-0.35
	proposed	-23.29	0	0.09

5. Conclusions

The characteristics of temporal-spatial correlation in MD are investigated, and a novel fast MD algorithm is proposed based on that analysis. By excluding some low-probability modes from the dynamic candidate mode list, 36.85% encoding time can be reduced on average, and the proposed algorithm can perform very close encoding efficiency compared with exhaustive MD algorithm.

6. References

[1] "Draft ITU-T Rec. and FDIS of Joint Video Spec. (H.264 | ISO/IEC 14496-10 AVC)", JVT of MPEG and VCEG, Doc. JVT-G050rl (2003).

[2] I.E.G. Richardson, *H.264 and MPEG-4 Video Compression, Video Coding for Next-generation Multimedia* (1st edition), John Wiley & Sons, England (2003).

[3] T. Wiegand, G.J. Sullivan, G. Bjntegaard and A. Luthra, "Overview of the H.264/AVC video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), pp. 560-576 (2003).

[4] C. Grecos and M.Y. Yang, "Fast inter mode prediction for P slices in the H264 video coding standard", *IEEE Transactions on Broadcasting*, 51(2), pp. 256-263 (2005).

[5] F. Pan, X. Lin, S. Rahardja, K.P. Lim, Z.G. Li, D. Wu and S. Wu, "Fast mode decision algorithm for intraprediction in H.264/AVC video coding", *IEEE Transactions on Circuits and Systems for Video Technology*, 15(7), pp. 813-822 (2005).

[6] J.F. Wang, J.C. Wang, J.T. Chen, A.C. T and A. Paul, "A novel fast algorithm for intra mode decision in H.264/AVC encoders", *Proceedings 38th IEEE ISCAS*, *Island of Kos 2006*, pp. 3495-3501 (2006).

[7] H. Li, Z.G. Li and C. Wen, "Fast mode decision algorithm for inter-frame coding in fully scalable video coding", *IEEE Transactions on Circuits and Systems for Video Technology*, 16(7), pp. 889-895 (2006).

[8] Y.C. Lin, T. Fink and E. Bellers, "Fast mode decision for H.264 based on rate-distortion cost estimation", *Proceedings 32nd IEEE ICASSP, Honolulu 2007*, pp. I-1137-1140 (2007).

[9] B. Zhan, B. Hou and R. Sotudeh, "Fast mode decision for inter-prediction in H.264/AVC", *Proceedings 7th IEEE ISCIT 2007*, *Sydney*, pp. 485-490 (2007).

[10] JVT reference software unofficial version JM10.1, http://iphome.hhi.de/suehring/tml/download