

Complexity Scalable Motion Estimation Based on Modes Pre-Selection in H.264

Dongming Zhang Chao Huang Shouxun Lin Yanfei Shen Lejun Yu

(Institute of Computing Technology of Chinese Academy of Sciences, Beijing China 100080)

Email: {dmzhang, chuang, sxlin, syf, jlyu}@ict.ac.cn

Abstract: H.264 allows tree-structured partitioning motion estimation performing on multiple reference frames. This new feature improves the prediction accuracy of inter-coding blocks significantly, but it is extremely computational intensive. The complexity of motion estimation increases linearly with the number of used reference frames and is in proportion to the number of modes. This disables some applications on computations-constrained platform. Complexity scalable algorithm is an algorithm that has several computation scales, each of which adapts to certain computation power platform. In this paper, we propose a complexity scalable algorithm for motion estimation in H.264 based on mode pre-selection in multiple reference frames. This proposed algorithm has 4 computation scales consisting of 72%, 47%, 33% and 20%. Simulation results show that this algorithm can effectively reduce complexity of motion estimation with gracefully degraded quality from the original.

Keywords: Complexity Scalable H.264
Modes Pre-Selection Motion Estimation

1. Introduction

H.264^[1] is the new video coding standard proposed by the JVT (Joint Video Team), which is aimed at high-quality coding of video contents at very low bit-rates. It uses the same hybrid block-based motion compensation and transform coding model as those existing standards, such as H.263^[2] and MPEG-2^[3]. Furthermore, a number of new features are introduced into H.264, such as multiple reference frames, sub-pixel motion estimation and tree-structured macroblock partitioning, to efficiently improve the encoding performance. They enable H.264 saves half of the bit-rate^[4] when compared with the H.263, and only uses about quarter of the bit-rate for the MPEG-2 at the cost of expensive computation and memory need.

Motion Estimation is the most important technique in inter-frame coding, and its computation occupies most of the whole video encoding. After tree-structured macroblock partitioning, multiple reference frames and sub-pixel motion etc. are introduced into motion estimation, the motion estimation already exceeds 85% computation of the encoding in reference software JM75C^[5] of H.264. In order to reduce the intensive computational requirements of motion estimation, many fast algorithms are put forward. E.g., Ting et al. proposed a center-biased reference frame pre-selection method to speed up motion estimation process which selects the "best" reference frame with minimum *SAD* (Sum of Absolute Differences) through checking only a few of searching points in all reference frames and applies full search in the selected reference frame^[6]. The method may lead to significant performance reduction especially in some sequences with much motion. Huang proposed a method to determine whether it is necessary to search in more reference frames via the available information after intra prediction and motion estimation from previous one reference frame^[7]. The two methods succeed in reducing the complexity of motion estimation to very low level, and video quality degradation is acceptable. However, in some applications, since their platforms can offer more computation power than the low complexity and they should get better video quality.

In this paper, we present a complexity scalable algorithm of motion estimation, which has four computation scales with different encoding performances and with which the more computation, the better video quality. The rest of this paper is organized as follows. In Section 2, we will analyze the distribution of the modes of macroblock among multiple reference frames. In Section 3, we will describe our complexity scalable motion estimation algorithm based on mode

pre-selection. Simulation results will be showed in Section 4. Finally, Section 5 gives a conclusion.

2. Analysis

In current H.264 reference software, search process is carried out reference frame by reference frame for all inter modes and direction by direction for all intra modes, and then the best mode is selected by minimized a Lagrangian cost function. The left of Fig. 1 shows the flow chart of the common search process. Let's assume that we have M block modes, N reference frames and that the search range for each reference frame and block mode is set $\pm W$ constantly, we need to check $M \times N \times (2W + 1)^2$ positions compared to only $(2W + 1)^2$ positions for a single reference frame and single block mode. This exhaustive search process is unacceptable especially on computation-constrained platform. In fact, most performance gain comes from a little of computation, while a lot of computation is wasted without any benefits. In the following, we will analyze the important factors to the complexity of motion estimation and mode decision.

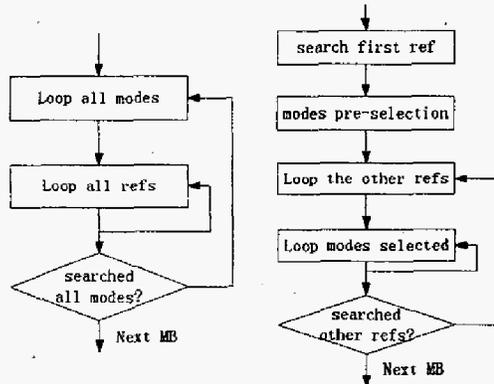


Fig. 1. Search process of one macroblock

At first, we will consider the number of reference frames. There are basically two types of temporal redundancy that can be captured by using multiple reference frames but not by traditional single frame. The first type of redundancy results from occluded and uncovered objects. Assuming that the current frame is frame t , sometimes some objects may be distorted or absent in frame $t-1$, but well represented in frame $t-2$ or frame $t-3$, etc. An example of which is the swing of bird's wings. The second type is sub-pixel movement of

textures. Because of the discrete nature of the capturing devices, textures and objects with different version of sub-pixel movement may occur in successive video frames. Referencing multiple frames leads to the complexity of search process increasing linearly with the number of reference frames.

Nevertheless the decrease of prediction residues with multiple reference frames depends on the natures of sequences but not on the number of searched frames. Sometimes the prediction gain is very significant, but sometimes, additional computation comes to naught. The distribution of reference frames determined by the reference software JM7.5C makes it evident. From Table 1 we can see that more than 90% of the optimal motion vectors selected by belong to the nearest reference frame (say the first reference frame, ref1). And the correlation between the current and the reference frame decrease with the temporal distance, e.g. for foreman sequence, 85.3% motion vectors come from ref1, 6.3% from the second reference frame (ref2), 3.7% from the third (ref3), 2.6% from the fourth (ref4) and 2.1% from the fifth (ref5). Since most of the motion vectors come from the first reference frame, we first apply full search for intra modes and inter modes from ref1.

Table 1. Distribution of optimal reference frame (%)

sequence	ref1	ref2	ref3	ref4	ref5
foreman	85.3	6.3	3.7	2.6	2.1
carphone	83.9	4.7	5.2	3.7	2.6
table	88.4	5.6	2.6	1.8	1.6
salesman	99.5	0.2	0.1	0.1	0.1
news	98.9	0.4	0.3	0.2	0.1
Average	91.2	3.4	2.4	1.7	1.3

The prediction gain of multiple coded modes comes from the variable textures of different macroblocks. Since a macroblock has a big size 16×16 , it is possible to contain more than one object that may move in different directions. Therefore more than one motion vector per macroblock may be needed to describe accurately the motion of all objects in it.

According to tree-structured macroblock partitioning, an inter macroblock may have modes $\text{inter}16 \times 16(\text{P}16 \times 16)$, $\text{inter}16 \times 8(\text{P}16 \times 8)$, $\text{inter}8 \times 16(\text{P}8 \times 16)$ and $\text{inter}8 \times 8(\text{P}8 \times 8)$, and when a macroblock select mode $\text{P}8 \times 8$, each 8×8 sub-block of it may further split to smallest size 4×4 , so it has shapes 8×8 , 8×4 , 4×8 and 4×4 . In addition, since one macroblock may contain

new objects, intra4x4 (I4x4) and intra16x16 (I16x16) modes are checked.

Table 2. Statistics of Selected Modes (%)

sequence	P16x16		P16x8		P8x16		P8x8		I4x4		I16x16	
foreman	45	14	11	29	13	22	29	14	1	35	1	4
carphone	53	11	9	28	11	27	22	17	2	39	3	12
stefan	36	22	10	24	6	35	43	9	4	45	2	16
news	76	2	3	14	5	15	15	5	1	25	0	4
salsman	82	1	2	8	2	16	14	3	0	21	0	0
Average	58	10	7	20	7	23	25	10	1	33	1	7

Now, we try to find the characteristic of the mode distribution. Since the multiple codes modes are aimed at various textures of macroblocks and the successive video frames usually have similar textures, the mode of one macroblock tends to be unchanged in different reference frame. In addition, the mode of one macroblock depends mainly on the first reference frame since their temporal distance is minimal. The experiment result verifies our analysis. In Table2, the left of each column, such as P16x16, is the percentage of the mode selected after searching the first reference frame and the right is the percentage of mode changed after searching the other four. We can see that in sequence news more than 75% macroblocks select P16x16 mode in the first reference frame, and only 2% macroblock select another modes after searching the other four reference frames. The sequence salesman has similar result. Even in the sequence stefan which has both local motion and global camera motion, 36%,10%,6%,43%, 4% and 2% of macroblocks are selected P16x16, P16x8, P8x16, P8x8, I4x4 and I16x16 mode respectively when only first frame is searched. After searching the other four, only correspondent 22%, 24%, 35%, 9%, 45% and 16% of macroblocks' modes changed. This means that $36\% \times 22\% + 10\% \times 24\% + 6\% \times 35\% + 43\% \times 9\% + 4\% \times 45\% + 2\% \times 16\% = 18.4\%$ of macroblocks' modes will change. That is to say 81.6% of macroblocks' modes will not be different between using the previous reference frame and using all five frames. Hence, we may search only the first reference frame in the case of computation-constrained applications and the video quality will not deteriorate too much.

Furthermore, we check a few candidate modes in more reference frames for better performance. In order to contain the best mode in the candidate modes as well

as possible, it is necessary to find the distribution of changed modes. Table 3 shows the changed modes' distribution of the sequence stefan and each row records the distribution of one changed mode, such as in first row, 36% of changed P16x16 came into P16x8, 29% into P8x16 and 35% into P8x8 after searching all frames. The other sequences have the similar distribution. We can see that (1) if the best mode after searching the first reference frame is inter mode, it is nearly impossible that the best mode after searching the others changed to intra modes; (2) if the best mode in previous frame is P16x8 or P8x16, it is possible that the global best mode belongs to P8x8; (3) when the best mode in previous frame is set to I4x4, it is likely to be changed to P8x8 if changed. And for I16x16, P16x16 is the best candidate.

Table 3. Distribution of changed modes of Stefan (%)

	P16x16	P16x8	P8x16	P8x8	I4x4	I16x16
P16x16	--	36	29	35	0	0
P16x8	28	--	17	55	0	0
P8x16	32	21	--	47	0	0
P8x8	31	39	30	--	0	0
I4x4	6	3	6	85	--	0
I16x16	56	24	16	4	0	--

3. Complexity scalable algorithm based on modes pre-selection

The search process of current reference software of H.264 doesn't exploit mode information in previous reference frame. We call it AMS (All Modes Search). In the following we will develop our complexity scalable motion search process using information about selected modes. In the process, we first check all candidate modes in previous reference frame for "best" mode and then pre-select candidate modes important to performance, which will be checked in the other four reference frames according to the "best" mode.

Now, we will define four methods of modes pre-selection. The first method to pre-select modes is defined as follow: if the best mode in previous frame is inter mode, both intra modes are disabled in the next reference frames; if it is I4x4 mode, only I4x4 and P8x8 are enabled; if it is I16x16, only I4x4 and P8x8 are disabled. The performance of the method will be nearly same to AMS, so we call it NNLS (Nearly No Loss Search). The second is LLS (Little Loss Search) in which if the best mode in previous frame is inter mode

or I4x4, the best mode and P8x8 are enabled for the next candidates; and if the best mode is intra16x16, the best mode and P16x16 mode are enabled. So there are one third of candidate modes for the next frames. The third, BMS (Best Mode Search) select the best mode in the first reference frame as the only candidate mode for the next reference frames. In the above three methods, all modes are candidates for the first reference frame. MMS (Main Modes Search) is the fourth method which checks only P16x16 and P8x8 modes for the first reference frame and then selects the better mode as the only candidate mode in the next reference frames.

From the definition of the methods of modes pre-selection, our algorithm will form four different computation scales of motion estimation below AMS's. The computation scale of NLLS depends on the nature of sequence, and it can be calculated approximately according to the distribution of modes in Table 2 as follow: $1/5 + 4/5 \times (98\% \times 4/6 + 1\% \times 2/6 + 1\% \times 4/6) = 72.8\%$. The scale of LLS should be $1/5 + 4/5 \times 2/6 = 46.7\%$. BMS's is $1/5 + 4/5 \times 1/6 = 33.3\%$ MMS has the lowest scale $1/5 \times 2/6 + 4/5 \times 1/6 = 20\%$.

Table 4. Performances using modes pre-selection (PSNR: dB, Bit-rate: kbps)

Sequence		foreman	carphone	stefan	news	salesman
AMS	PSNR	35.88	37.19	34.39	36.73	35.58
	Bit-rate	80.68	74.63	283.75	54.79	38.53
NLLS	PSNR	35.89	37.21	34.38	36.74	35.58
	Bit-rate	80.84	74.77	282.85	54.80	38.53
LLS	PSNR	35.80	37.14	34.37	36.72	35.58
	Bit-rate	80.60	74.78	283.21	54.92	38.60
BMS	PSNR	35.79	37.11	34.35	36.71	35.56
	Bit-rate	81.64	75.72	282.40	55.12	38.55
MMS	PSNR	35.76	37.04	34.31	36.65	35.52
	Bit-rate	82.55	75.29	278.86	56.48	39.32

4. Simulation Results

For integration our algorithm into H.264, we adapted the structure of JM75C and the new search process is showed in the right of Fig 1. In our simulations, we choose five sequences (QCIF) to check the performance of the four mode pre-selection method and AMS among which foreman, carphone and stefan have both local motion and global camera motion, while news and salesman only have little local motion. Each sequence is encoded 105 frames at 30fps, CABAC entropy coder, QP of 28, search range of 16 and 5 reference frames.

In Table 4, we can see that NLLS's performance is best and nearly same to AMS's. The performance of LLS and BMS continues decreasing, and the average quality degradation is about 0.02dB and 0.04dB respectively. MMS's performance is the lowest, and the average degradation is about 0.1dB.

5. Conclusion

In this paper, we propose a complexity scalable motion estimation based on mode pre-selection, and the complexity of motion estimation has four scales: 72%, 46%, 33%, and 20%. Simulation results show that our four methods can get graceful quality degradation with computation scale's decreasing.

References

1. "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T, JVT050, 2003.
2. "Video Coding for Low Bit Rate Communication," ITU-T Recommendation H.263 version 1, 1995.
3. "Generic Coding of Moving Pictures and Associated Audio Information -Part 2: Video," ITU-T and ISO/IEC JTC 1, ITU-T Recommendation H.262 and ISO/IEC 13 818-2(MPEG-2), 1994.
4. T. Wiegand, J. Gary, G. Bjøntegaard et al., "Overview of the H.264/AVC coding standard," IEEE Trans. Circuit Syst. Video Tech., vol.13, pp.560-575, Jul. 2003.
5. JVT reference software JM75C, <http://bs.hhi.de/~suehring/tml/download/jm75c.zip>
6. C. W. Ting, L. M. Po and C. H. Cheung, "Center-biased frame selection algorithms for fast multi-frame motion estimation in H.264," Proceeding of 2003 IEEE International Conference on Neural Networks and Signal Processing, pp. 1258-1261, Dec. 2003.
7. Y. W. Huang, B. Y. Hsieh, T. C. Wang, et al. "Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264," Proceedings of 2003 IEEE International Conference on Acoustics Speech, and Signal Processing, vol. 2, pp. 809-812, Jul. 2003.