FAST INTER FRAME ENCODING BASED ON MODES PRE-DECISION IN H.264

Dongming Zhang[1, 2] *Yanfei Shen*[1] *Shouxun Lin*[1] *Yongdong Zhang*[1]

1 Institute of Computing Technology, Chinese Academy of Sciences({dmzhang,syf,sxlin,zhyd }@ict.ac.cn)
2 Graduate School of the Chinese Academy of Sciences

ABSTRACT

The new video coding standard, H.264 allows motion estimation performing on tree-structured block partitioning and multiple reference frames. This feature improves the prediction accuracy significantly, but the cost of which is the complexity and computation load of video coding increase drastically. The reference software of H.264 adopts a full search scheme and its complexity increases linearly with specified the number of reference frames and the number of modes respectively. In this paper, we propose a fast mode decision algorithm in H.264 which eliminates searching unnecessary modes. The proposed algorithm takes full use of available information obtained from the previous searching process, e.g., macroblock's best mode in searching the previous reference frames. In the algorithm implementation, multiple half-stop conditions have been set in motion estimation so as to decrease encoder's complexity. Simulation results show that this proposed algorithm can effectively reduce complexity of inter-frame encoding, and the quality degradation is tiny compared with full search scheme. Furthermore, our algorithm is adaptive to variant test sequences, no need to set specific experimental threshold.

1. INTRODUCTION

H.264 [1] is a new video coding standard proposed by the JVT (Joint Video Team). 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]. At the mean time, many 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. As a result, H.264 can save half of the bit-rates [4] when compared with the H.263, but the cost of high coding performance is intensive computation and expensive memory need.

In inter-frame coding, mode decision includes motion estimation and intra predication and consumes the heaviest computation. When tree-structured macroblock partitioning, multiple reference frames and sub-pixel motion are introduced into motion estimation, the computation of motion estimation already exceeds 85% of the whole encoding in JM73 [5]. In the existing reference software of H.264, mode decision is carried out reference frame by reference frame for all inter modes, and 13 intra prediction modes (9 for I4x4 and 4 for I16x16) are checked. The left of Fig. 1 shows the flow chart of the full search process.

Let's assume that we have M block modes, N reference frames and that the search range is $\pm W$ constantly, we need to check $M \cdot N \cdot (2W+1)^2$ 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. However, in fact, a lot of computation is wasted without any benefits. In order to eliminate extra mode decision, many fast algorithms are put forward. Much literature contribute to decrease the check points via elimination unnecessary points from $(2W+1)^2$ points such as classical fast motion estimation algorithm TSS(three-step search)[6]. Furthermore, in H.264 block mode M and reference frame number N are important factors to the complexity of motion estimation. Ting et al. proposed a center-biased reference frame pre-selection method to speed up motion estimation process which firstly selects the best reference frame with minimum SAD (Sum of Absolute Differences) through checking a few of searching points in all reference frames, instead of all points in the specified search range and then applies full search in the best reference frame [7]. The method leads to significant performance reduction for some sequences. The main reason lies in that the method treats all reference frames uniformly (but in fact, the importance of each reference frame to performance is decreasing from the nearest reference frame to the furthest) and the firstly checked points can't describe the motion in

0-7803-9332-5/05/$20.00 ©2005 IEEE

sequence well. Huang proposed a method to determine whether it is necessary to search the rest four reference frames via the available information after intra prediction and motion estimation from previous one reference frame [8]. This method succeeds in reducing the complexity of motion estimation to very low level with acceptable video quality degradation. But the threshold for intra mode introduced must be adjusted manually according to the content of different sequence. At the mean time, when the condition for search stop does not meet, the all the rest 4 frames will be searched.

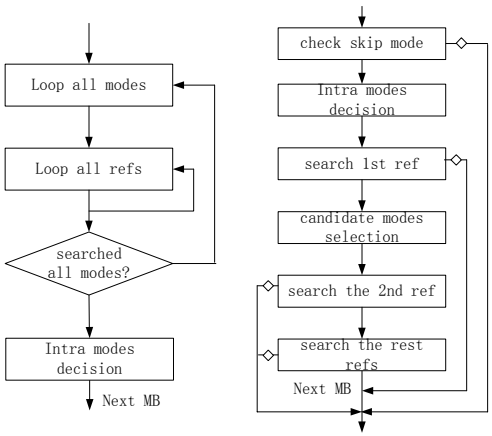

Fig. 1. Search process of one macroblock in inter frame encoding (◇ indicate a logical operation, if result is true, search will stop)

In this paper, we propose a fast algorithm based on modes pre-decision. The algorithm includes checking SKIP mode, intra mode and reducing modes gradually with the increasing of distance of reference frames via available information provided by the previously searched reference frames. 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 fast algorithm based on mode pre-selection. Simulation results will be showed in Section 4. Finally, Section 5 gives a conclusion.

2. OBSERVATION

The common ground of the two fast searching methods proposed by literature [7] and [8] lies in eliminating searching unnecessary reference frames, i.e., decreasing *M*. Decimating unnecessary modes is another important way to make mode decision quickly. In our proposed algorithm, we will combine elimination of reference frames and modes and we will integrate halfway-stop technique. In the following, we will give many observations which will provide cues to select reference frames and modes.

At first, we will consider the selection of reference frames. 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 decreases with the temporal distance, e.g. in foreman sequence, 85.3% motion vectors points to ref1, 6.3% to the second reference frame (ref2), 3.7% to the third (ref3), 2.6% to the fourth (ref4) and 2.1% to the fifth (ref5). Since most of the motion vectors point to the first reference frame, we first apply full search for all inter modes in ref1.

Then, we will consider the decision of block size. The prediction gain of multiple bock sizes comes from the variable textures or motion of different macroblocks. Since one macroblock may contain more than one object that may move in different directions, more than one motion vector may be needed to describe accurately the motion of all objects.

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

According to tree-structured macroblock partitioning, an inter macroblock may have modes Inter16x16(P16x16), Inter16x8(P16x8), Inter8x16(P8x16) and Inter8x8(P8x8), and when a macroblock selects mode P8x8, each 8x8 sub-block may further split into smaller size, and 4x4 is the smallest shape, so it has shapes 8x8, 8x4, 4x8 and 4x4. In addition, since one macroblock may contain new objects or uncovered objects, Intra4x4 (I4x4) and Intra16x16 (I16x16) modes are checked.

Now, we try to find the characteristic of the mode distribution. Since the successive video frames usually have similar textures, the best mode of one macroblock tends to be independent of different reference frame. In addition, the best mode of one macroblock depends mainly on the first reference frame because their temporal distance is minimal. These data in Table2 verify our analysis. The left of each column is the percentage of the selected mode after searching ref1 and the right is the percentage of the changed mode after searching the other four. We can see that in sequence news more than 75% macroblocks select P16x16 mode in ref1, and only 2% macroblock select another modes after searching the rest. The other sequences have very similar results besides stefan. Stefan has both local motion and global camera motion, 36%,10%,6%,43%, 4% and 2% of macroblocks are selected as P16x16, P16x8, P8x16, P8x8, I4x4 and I16x16 respectively when ref1 is searched, and after searching the rest, correspondent 22%, 24%, 35%,9%, 45% and 16% of macroblocks' modes changed. This

Table 2. Distribution of Best Modes of macroblock in inter frames(%)

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

means that 36%×22%+10%×24%+6%×35%+43%×9%+ 4%×45%+2%×16%=18.4% of macroblocks' modes will change. So if we only search the first reference frame to select the best mode, the encoder may miss real best mode and this may deteriorate video quality badly. To avoid this happen, after search the first reference frame, some most probable modes should be selected as candidate modes for the next reference frames according to the best mode.

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	--

In order to include the real best mode in the candidate modes as well as possible, it is necessary to find the distribution of changed modes. Table3 shows the changed modes' distribution of the sequence stefan and each row records the distribution of one changed mode, such as first row, 36% of changed P16x16 came into P16x8, 29% into P8x16 and 35% into P8x8 after searching all frames. Because little percentage of macroblocks will change modes (referring to Table2) and because of their low or smart motion, the other sequences' distribution of changed modes fits the following five rules better than stefan, so the detailed data are not listed.

(1) If the mode is P16x16, the real best mode comes to P16x16 with much possible;

(2) If the mode is P16x8, P16x8 and P16x16 become the real best mode more likely than P8x16;

(3) If the mode is P8x16, P8x16 and P16x16 become the real best mode more likely than P16x8;

(4) If the better intra mode is I4x4, the most probable inter mode is P8x8;

(5) If the better intra mode is I16x16, P16x16 is the most probable inter mode.

3. FAST MODES DECISION ALGORITHM

The search process of current reference software of H.264 doesn't exploit any mode information in previous reference frame. We call it AMS (All Modes Search).

In the following we will set up our fast algorithm for mode and reference frame decision according to observations described above.

In order to find the most probable mode of one macroblock in inter picture as soon as possible, we introduce one adaptive threshold to check SKIP macroblock and half-way stop technique. The detail processing is as follow:

Step1: Check SKIP mode in the first reference frame, check halfway-stop condition I, if the condition meets, set best mode as SKIP mode and go to step7.

Step2: Check I4x4 and I16x16 and record the better intra mode and cost as *BetterIntraMode* and *CostIntraMode* respectively.

Step3: Check all inter modes in first reference frame to find the best inter mode and compare it with the better intra mode to find the best mode. If the best mode is *BetterIntraMode* or P16x16, go to step7; otherwise, record the best mode and the cost as *BestMode* and *CostBestMode* respectively.

Step4: Pre-select candidate modes for the second reference frame.

```
if(BestMode == P16x8)
{
   select P16x8 and P8x8 as candidate modes;
   if(BetterIntraMode = =I4x4)
      split each 8x8 block;
}
else if(BestMode == P8x16)
{
   select P8x16 and P8x8 as candidate modes;
   if(BetterIntraMode = =I4x4)
     split each 8x8 block;
}
else{
   select P8x8 as candidate mode;
   if(BetterIntraMode = =I16x16)
    select P16x16 as candidate mode;
}
```

Step5: Check candidate modes in the second reference frame and find the best mode. Then check halfway-stop condition II. If condition meets, go to step7; otherwise, renew the *BestMode* and *CostBestMode*.

Step6: Check *BestMode* in the next reference frame and check the half-way stop II, if condition meets, stop searching, otherwise loop Step6 until searching all reference frames.

Step7: Output the best mode and best reference frame (if necessary).

In step1, the half-way stop I condition is that the distortion with zero motion vectors is less than *Tskip*. The *Tskip* is an adaptive threshold, and it will be set as a small enough value at the initial stage and it will be renewed by the distortion of real SKIP macroblocks, $T_{skip} = (Distorn + T'_{skip})/2$, where T'_{skip} is the current used SKIP mode threshold and *Distorn* is the SAD of the macroblock just encoded as SKIP.

In step5 and step6, the half-way stop II is that the cost of encoding macroblock with current best mode and current reference frame is not less than *CostBestMode,* the cost of previous reference frame.

So in our proposed algorithm, no experimentally threshold is needed.

4. SIMULATION RESULTS

For integration our algorithm into H.264, we adapted the structure of JM73. The new search process is showed in the right of Fig 1.

Table 4. Encoding performance of the proposed algorithm and JM73 (PSNR: dB, Bit-rate: kbps)

sequence		foreman	carphone	stefan	news	salesman
Prop. 5 refs	PSNR	35.74	37.25	34.28	36.72	35.52
	Bit-rate	80.96	69.12	358.40	55.36	39.84
JM73 5 refs	PSNR	35.85	37.35	34.35	36.72	35.52
	Bit-rate	80.00	67.36	340.80	54.40	39.84
JM73 1 ref	PSNR	35.53	36.92	34.19	36.69	35.50
	Bit-rate	86.90	74.40	355.20	54.24	39.68

Table 5. Consuming time per macroblock of the proposed algorithm and JM73(ms)

sequence	foreman	carphone	stefan	news	salesman
Prop. 5 refs	4.2	3.8	8.5	1.7	1.9
JM73 5 refs	22.5	19.6	31.3	17.6	17.9
JM73 1 ref	4.6	4.1	6.2	3.6	3.5

In our simulations, we choose five sequences (QCIF) to check the performance of the proposed fast algorithm. Among which foreman, carphone and stefan have both local motion and global camera motion, while news and salesman only have little local motion. Table4 lists the encoding performance of our proposed algorithm with 5 reference frames, JM7.3 with 5 reference frames and JM7.3 with one reference frame. And the other encoding parameters are set as follow: entropy coding using CABAC, using RDO, QP is set 28, search range is 16, all coded modes are enabled. Table 5 lists the consuming time per macroblock of our proposed algorithm with 5 reference frames, JM7.3 with 5 reference frames and JM7.3 with one reference frame. The platform CPU is Pentium IV 2.4 GHz.

From Table 4, we can see the performance of our proposed algorithm is very close to JM73 using 5 reference frames and much higher than JM73 with one reference frame. And from Table 5, we can see that the consumed time by JM73 with 5 reference frames is about 4~5 times than JM73 with one reference frame, while our proposed algorithm with 5 reference frames consumes less time than JM73 with 1 reference frame except stefan, in which the complex motion and variable texture are two main reasons.

5. CONCLUSION

In this paper, we propose a fast algorithm based on modes pre-selection, and it can reduce the complexity of video encoding compared with the JM73's, but the reduction percentage varies from different sequence. On the other hand, simulation results show that the proposed fast algorithm does not deteriorate the quality of video significantly. Furthermore, because no experimentally value is introduced into it, our proposed algorithm is adaptive and can obtain comparatively good coding performance independent of sequence.

This work is supported by National Nature Science Foundation of China under grant number 60302028.

6. 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 JM73, http://bs.hhi.de/~suehring/tml/download/JM73.zip

[6]T. Koga, K. Iinuma, A. Hirano, et al. , " Motion-compensated interframe coding for video conferencing," Proceeding NTC 81, pp. C.9.6.1-9.6.5, New Orleans, LA, Nov./Dec. 1981

[7]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.

[8]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.