

FAST MODE DECISION FOR INTER PREDICTION IN H.264¹

Qionghai Dai ,Dongdong Zhu and Rong Ding

Broadband Networks & Digital Media Lab of Automation Dept

Tsinghua University, Beijing 100084, China

qhdai@mail.tsinghua.edu.cn

ABSTRACT

This paper proposes a fast inter prediction mode decision method for H.264. By pre-encoding a down-sampled small image, the candidate inter block modes can be reduced to a small subset. The simulation result shows that our algorithm can achieve up to 50% complexity reduction with less than 0.2dB PSNR decrease.

1. INTRODUCTION

JVT/AVC/H.264 is the newest international video coding standard approved by ITU-T and ISO/IEC. It shows significant performance gain over contemporary video coding standards. When compared with MPEG2, H.264 can save nearly 50% of bit rate under same visual quality [2].

The performance improvement is at the expense of computational complexity increment. H.264 main profile decoder requires 3 times computing capability than MPEG4 simple profile, and an increase of more than one order of magnitude of the computational complexity is reported at encoder end [3].

H.264 defines 7 block modes for inter prediction (figure 1), 9 modes for 4x4 intra prediction and 4 modes for 16x16 intra prediction. During the encoding process, the encoder has to examine all these modes to achieve the best prediction. When RDO is enabled, each prediction mode will induce a real encoding process, which includes motion estimation (ME), transformation, quantization and entropy coding for inter prediction modes and intra prediction, transformation, quantization and entropy coding for intra prediction modes. This mode selection process results in significant performance improvement as well as heavy computational load at the encoder. If we can limit the candidate prediction modes to a small subset, the computational complexity could be reduced.

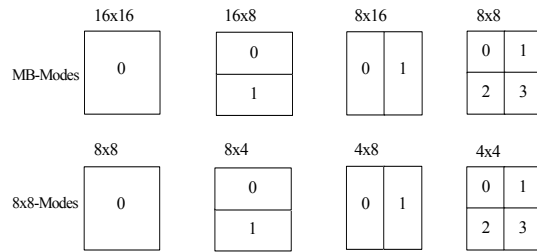


Fig1. Block modes for inter prediction

F. Pan proposed a fast intra prediction selection algorithm in [4], which can speed up the encoding process with negligible PSNR loss and bit rate increase.

However, the selection of inter prediction mode would be much more complex than that of intra prediction, since the block size used for motion estimation has relations to not only the image being encoded but also the reference image. When considering R-D optimization, this relationship will become more obscure and make the pre-choice of inter prediction mode almost impossible. In this paper, we present a novel fast inter prediction mode decision algorithm which can limit the candidate modes to a small subset by pre-encoding a down-sampled small image. The simulation result shows that our algorithm can reduce nearly 50% of encoding time.

This paper is organized as follows. Section 2 gives a description of our algorithm. Section 3 shows the simulation results for several test sequences. Section 4 provides some concluding remarks.

2. FAST INTER MODE DECISION

The encoder structure of this algorithm is shown in figure 2:

¹ This work is supported by the National Natural Science Foundation of China (No. 60172026) and by the Basic Research Foundation of Tsinghua University (No. JC2001028)

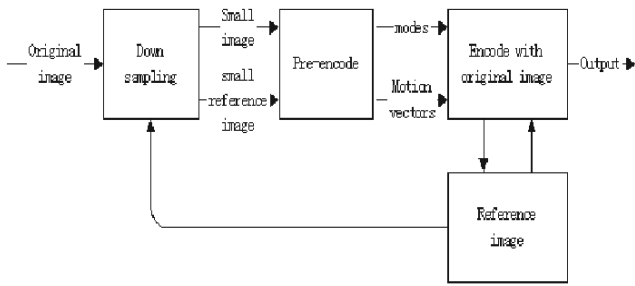


Fig2. Framework of the fast inter prediction mode decision algorithm

Firstly, we down sample the original image to a small image of 1/2 low resolution. Then pre-encode the small image with prediction mode selection method used in JM61 [8] and get the prediction mode of each 8x8 block. When encoding each 16x16 macro block of the original image, we map the prediction mode of collocated 8x8 block in the small image to a candidate modes subset and use motion estimation and R-D optimization to choose the best mode from the subset. Because the pre-encode process is performed on a 1/2 low-resolution small image, the encoding time used is about 1/4 that of the original size image. After pre-encoding, the candidate modes subset used for encoding original image become very small, and with this candidate modes subset, the encoding time will be reduced to 1/4 of the original. Thus the overall encoding speed will be doubled.

2.1 Pre-Encoding

During the encoding process of inter prediction image (P or B), we firstly use 7-tap filter (-1,0,9,16,9,0,-1) on original image and reference image respectively to get down-sampled 1/2 low resolution small images. After down-sampling, if the width or height of the small image is not a multiple of 16, extend to a multiple of 16 by using pixels on the edge.

The pre-encoding process is about the same as the encoding process of [8] except for a few differences:

- In [8] R-D optimization is performed on macro block level. Considering the spatial relationship of 2:1 between original image and down-sampled image, the R-D optimization during pre-encoding process needs to be done on 8x8 block level. This means we do not use 16x16, 16x8 and 8x16 blocks in pre-encoding process.
- During the pre-encoding process, motion search range should be scaled to 1/2 of the original one.

- In case of the best prediction mode mismatch between pre-encoding process and real encoding process caused by information loss during down-sampling, instead of choosing only the best mode during R-D optimization, the pre-encoding process should store the top 2 best modes and pass them to real encoding process to build the candidate modes subset.

The pre-encoding process passes the top 2 best modes and all motion vectors gotten under these modes to the real encoding process.

2.2 Encoding

During the real encoding process of original size image, for each 16x16 macro block, the top 2 best prediction modes of the collocate 8x8 block in down-sampled small image are mapped to a candidate modes subset:

Table 1. Mapping of modes obtained from pre-encoding process to a candidate modes subset

Modes obtained from pre-encoding process		Candidate modes
Macro block mode	8x8 sub-partition mode	
SKIP/DIRECT		SKIP/DIRECT , 16x16
116x16		116x16
14x4		14x4, 116x16
P8x8	DIRECT	DIRECT , 16x16
	8x8	8x8, 16x16
	8x4	8x4, 16x8
	4x8	4x8, 8x16
	4x4	See below

The smallest block size in H.264 is 4x4. Due to the 1:2 spatial relationship between down-sampled small image and original image, 8x4, 4x8 and 4x4 blocks in original image have no matched block modes in the small image. The blocks smaller than 8x8 are mainly used for encoding regions containing complex motion which is very likely due to this region containing many small objects moving independently. Using small blocks can separate these objects into independent blocks and perform motion estimation separately. Based on this fact, we use objects distribution information to help us to choose block modes which has the block size smaller than 8x8.

For the sake of simplicity, we use the algorithm described in [4] to get the objects edges. Firstly, the Sobel operator [6] is applied on the macro block and

an edge vector is generated for each pixel in the macro block.

$$dx_{i,j} = p_{i,j+1} + 2 \times p_{i,j} + p_{i,j-1} - p_{i,j+1} - 2 \times p_{i,j} - p_{i,j-1} \quad (1)$$

$$dy_{i,j} = p_{i+1,j} + 2 \times p_{i,j} + p_{i-1,j} - p_{i+1,j} - 2 \times p_{i,j} - p_{i-1,j}$$

Define:

$$Amp_{i,j} = |dx_{i,j}| + |dy_{i,j}| \quad (2)$$

$$Ang_{i,j} = \frac{180}{\pi} \times \arctan \left(\frac{dy_{i,j}}{dx_{i,j}} \right) \quad (3)$$

Build edge direction histogram for horizontal and vertical direction.

$$Histo(k) = \sum_{(i,j) \in SET(k)} Amp_{i,j} \quad (4)$$

$$SET(k) \in \{ \{i_1, j_1\}, \dots, \{i_u, j_u\}, \dots, \{i_3, j_3\} \mid Ang_{i,j} \in a_k \},$$

$$a_1 = [-22.25^\circ, 22.25^\circ]$$

$$a_2 = (-90^\circ, -67.5^\circ) \cup (67.5^\circ, +90^\circ)$$

$$a_3 = \Omega - (a_1 \cup a_2)$$

k=1 for horizontal direction, 2 for vertical direction and 3 for other.

We choose block modes according to object edge direction:

```

if(Histo(1)>2xHisto(2) and Histo(1) > 2xHisto(3))
    choose 8x4 block mode
else if(Histo(2)>2xHisto(1) and Histo(2) > 2xHisto(3))
    choose 4x8 block mode
else
    choose 4x4 and 8x8 block modes

```

The motion vectors obtained during the pre-encode process can also be used to make the motion estimation search center more accurate. We use MVFAST motion estimation algorithm described in [5] for motion search. In MVFAST motion estimation algorithm, an initial motion vector set S needs to be built which includes vector (0,0) and motion vectors of neighboring blocks. Before diamond pattern search, motion needs to be classified as low, medium and high:

$$Vi(x_i, y_i) \in \text{initial motion vector set} \quad (5)$$

define :

$$Li = |x_i| + |y_i|$$

$$L = \max(Li)$$

$$\text{motion} = \begin{cases} \text{low}, & \text{if } L \leq TH_1 \\ \text{medium}, & \text{if } TH_1 < L \leq TH_2 \\ \text{high}, & \text{if } L > TH_2 \end{cases}$$

If motion=low, small diamond pattern is chosen for motion search.

If motion=medium, large diamond pattern is chosen first for motion search, and then to be refined by using small one.

If motion=large, a test must be performed on all initial vectors, use the vector which yield the smallest

SAD as search center, then use small diamond pattern to perform motion search.

In our algorithm, we build two initial motion vector set: S1 includes vector (0,0) and motion vectors of neighboring blocks; S2 includes the scaled version of motion vectors obtained from pre-encoding process. Firstly, an early termination test is performed on S2. For each initial motion vector in S2, SAD is calculated and min SAD is gotten as:

$$SAD_{\min} = \min(SAD(V_i)) \quad (6)$$

$$Vi(x_i, y_i) \in S_2$$

If SAD_{\min} is smaller than a threshold T_1 we may terminate the motion estimation process immediately. In this case the initial motion vector in S2 which generates the min SAD is selected as the final integer motion vector for this block type. T_1 can be set equal to the number of pixels of the examined block type.

If the early termination condition is not satisfied, the S2 and S1 are combined into one initial motion vector set S, and a normal MVFAST motion estimation process is performed.

In our algorithm, complexity reduction is achieved on two levels:

(1) On mode selection level, if the collocated 8x8 block in down-sampled image is encoded using inter prediction mode, no intra prediction modes will be examined during the real encoding process. And in most case, only 4 inter prediction modes will be tested.

(2) On motion search level, using the motion vectors obtained during the pre-encode process can make the motion estimation search center more accurate, thus fewer search steps are needed.

3. SIMULATION RESULTS

Our algorithm is implemented on reference software JM61 provided by ITU-T. To speed up the test process, we optimized JM61 with MVFAST motion estimation algorithm [5] and fast intra mode decision algorithm [4]. In the following sections, we call the JM61 without our algorithm as JM_OLD (optimized using MVFAST and Fast Intra mode decision) and JM61 with our algorithm as JM_NEW (optimized using MVFAST, Fast Intra mode decision and our algorithm).

Although manually setting the prediction mode number in configuration file to only one or two modes for JM_OLD can also save encoding time, significant performance loss might be introduced. So for JM_OLD, we perform the test for 2 configurations respectively: 1) with all 7 inter block modes and all intra prediction modes; 2) with only 16x16 inter block mode and all intra prediction modes.

Three standard test sequences have been tested with QP from 20~40. The testing environment includes a P4 1.5GHz PC with 256M memory and Windows 2000 OS.

Table 2. Simulation result when comparing JM_NEW with JM_OLD

Sequence	Max PSNR Change(db)		Avg Time Saving(%)	
	Config1	Config2	Config1	Config2
Stefan	-0.01	0.91	42	2
Foreman	-0.18	0.73	47	-1
Paris	-0.18	1.01	53	0

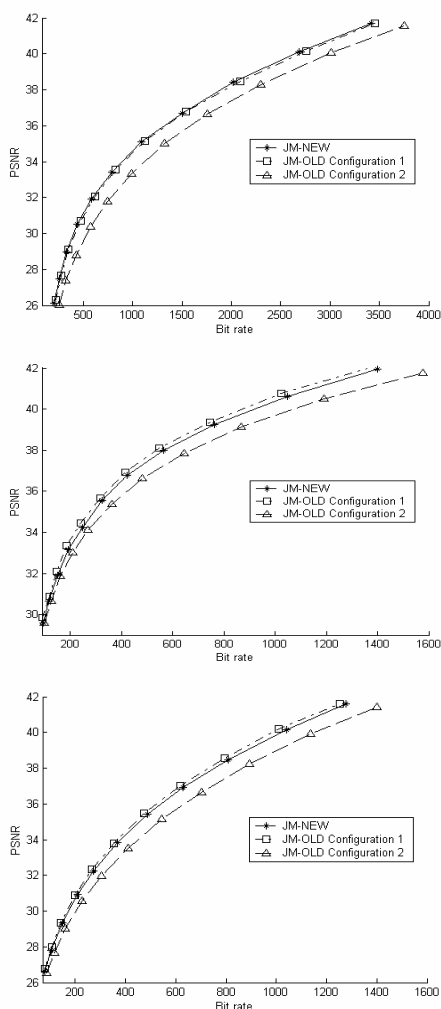


Fig3. PSNR vs. bit-rate for Stefan, Foreman and Paris respectively

When compared with JM_OLD configuration 1, our algorithm doubles the encoding speed with negligible loss of PSNR. When compared with JM_OLD configuration 2, our algorithm has a significant advantage on performance.

4. CONCLUSION

In this paper, a novel fast inter prediction mode decision algorithm for H.264 is presented. A down sampled small image of original image is pre-encoded to get best inter prediction modes in the small image. Then the modes obtained from the pre-encoding process are mapped to a candidate modes subset used for original image encoding process. The motion vectors obtained from pre-encoding process are also used to make motion estimation search center more accurate.

The simulation result shows that our algorithm can reduce nearly 50% of encoding time with PSNR decrease less than 0.2dB.

5. REFERENCES

- [1] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification", JVT Doc JVT-G050, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6) 7th Meeting, Pattaya, Thailand, 7-14 March, 2003.
- [2] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, Ajay Luthra, "Overview of the H.264/AVC video coding standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003.
- [3] Massimo Ravasi, Marco Mattavelli, Christophe Clerc "A Computational Complexity Comparison of MPEG4 and JVT Codecs", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6) 4th Meeting, Klagenfurt, Austria, 22-26 July, 2002.
- [4] F. Pan, X. Lin, R. Susanto, K. P. Lim, Z. G. Li, G. N. Feng, D. J. Wu, and S. Wu, "Fast Mode Decision Algorithm for Intra Prediction in JVT", G013, 64th MPEG and 7th JVT meeting, Pattaya, March 2003.
- [5] P.I. Hosur and K.K. Ma, "Motion Vector Field Adaptive Fast Motion Estimation," Second International Conference on Information, Communications and Signal Processing (ICICS '99), Singapore, 7-10 Dec '99.
- [6] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing," Prentice Hall, 2002
- [7] Sang Uk Lee, Jong Won Kim, "Variable Block Size Techniques for Motion Sequence Coding"
- [8] ITU-T H.264 reference software,"JM61e"
- [9] S. Zhu and K.K. Ma, "A new diamond search algorithm for fast block matching motion estimation," Proc. of Int. Conf. Information, Communications and Signal Processing, vol.1, pp.292-6, 1997.