FAST MOTION ESTIMATION FOR MPEG-2 TO H.264 TRANSCODING

Gao Chen [1,2] Shou-xun Lin [1] Yong-dong Zhang [1] Feng Dai [1,2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[2] Graduate School of the Chinese Academy of Sciences, Beijing, 100039, China

Abstract

To balance the rate-distortion (R-D) performance and the computational cost of the MPEG-2 to H.264 transcoder, we propose a new intelligent algorithm to select macroblock (MB) mode and decide the search window size based on the energy of MPEG-2 residual MB. The proposed method remarkably reduces the transcoding complexity, and achieves similar coding gain compared to the MPEG-2 to H.264 transcoder using exhaustively full search method to determine the best block size mode. The high complexity reduction at very small rate losses makes the proposed algorithm is helpful in real-time implementation of the MPEG-2 to H.264 video transcoder.

Keywords: Transcoding, H.264, MPEG-2, Mode Decision.

1. Introduction

As a result of the potential of the H.264 [1] to replace MPEG-2 video in digital video systems, there is a strong need for research into transcoding technologies to enable smooth transition from the widely available MPEG-2 video to H.264 format and vice versa. Inherent from high complexity of H.264 coding algorithm, the MPEG-2 to H.264 transcoding is much more complex compared to other video format transcoding. So, how to reduce the transcoding computational complexity while maintaining properly visual quality is the highly concerned issue [2].

Due to the high computational complexity of the H.264, the H.264 re-encoding process is the most critical part in the MPEG-2 to H.264 transcoder. Furthermore, in a hybrid-coding video encoder, most of the computation is spent on motion estimation (ME), especially for H.264. The results of H.264 complexity analysis listed in the literature [3] show that the most time consuming module in H.264 encoder is ME which occupies over 80% of computation resource. So, optimization of ME will greatly reduce the overall complexity of transcoding process. In this paper, we focus on how to exploit the motion information carried in the inputting MPEG-2 bit-streams to reduce the complexity of H.264 re-encoding process in MPEG-2 to H.264 transcoder. All the work is discussed within P (inter) frames, while this framework can be easily extended to B (bi-predictive) frames.

The rest of the paper will be organized as follows: Section 2 provides some observation and analysis on the energy of MPEG-2 residual MB. Section 3 describes the proposed fast multi-block selection and adaptive search window size choosing algorithms. The experimental results are presented in section 4. We close the paper with concluding remarks in section 5.

2. Observations on the energy of MPEG-2 residual MB

The direct way to transcode video is cascading a MPEG-2 video decoder with a H.264 video encoder, i.e. the cascaded pixel-domain transcoder (CPDT). Obviously, this direct approach is highly complex because it encompasses the both complexity of the full MPEG-2 decoding and full H.264 encoding. It is also called the reference architecture as it represents the upper bound on the R-D performance of the transcoded video [2]. In this paper, we adopt a partial re-encoding architecture for MPEG-2 to H.264 transcoder, as shown in Figure 1, which consists of two stages: a MPEG-2 decoding stage followed by a H.264 partial re-encoding stage which use the retrieved motion information gathered in the first stage to reduce the computational cycles taken by ME.

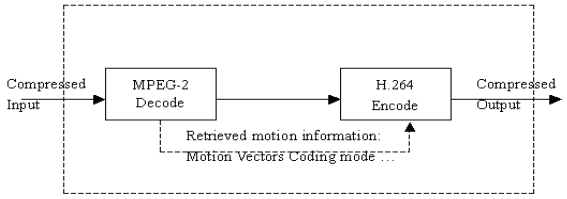


Fig 1. MPEG-2 to H.264 transcoder

In H.264 codec, the center of motion search window (MSW) is located at the position pointed by the predicted MV [1]. In MPEG-2 to H.264 transcoding, the MPEG-2 MVs, are the results of MPEG-2 ME, can be used as the center of MSW instead. The experimental results, which are omitted in this paper, prove that is efficient. So, the center of

MSW we use hereafter is the position pointed by the MPEG-2 MV.

As the new feature of variable-sizes block-matching (VSBM) in H.264 allows seven different block-sizes in ME is computational expensive, how many block-sizes is suitable in transcoding is a trade-off between performance and complexity. Table 1 shows our empirical results in transcoding ten different standard test video sequences from MPEG-2 to H.264 using the CPDT architecture with only four block-sizes (mode 1 to 4), and with all seven block-sizes.

Sequences	△PSNR (dB)	△bit-rate (%)
Coastguard	0.030	- 1.38
Flower	0.038	- 0.53
Foreman	0.026	0.28
Kiel	0.034	- 1.52
Mother & Daughter	0.002	0.08
News	0.022	0.41
Silent	0.014	- 0.04
Singer	0.034	0.99
Stefan	0.050	- 0.67
Template	0.042	- 1.50

Table 1. Performance of the MPEG-2 to H.264 transcoder employing only four modes

To simplify our comparison, we have used average PSNR gain (△PSNR) and bit-rate reduction (△bit-rate) results, based on five different quantizers (40, 36, 32, 28, 24). Compared to using only four block-sizes, an average of 0.029 dB and －0.39% bit-rate reduction can be obtained by using all seven block-sizes. The achievable performance improvement is insignificant in our simulations while using all seven block-sizes is time-consuming. To strike a good balance of transcoding performance and computational complexity, we only enable four block-sizes (mode 1 to 4) in the following discussion.

It is well known that the performance of inter mode can be measured by the energy of residue. The measure of the energy we used is the sum of the absolute value of the dequantized DCT coefficients of the motion compensated prediction MPEG-2 residual MB. Let the variable E be defined as the energy of the MPEG-2 residual MB, the function $Prob_1$（$V_1 \leqslant E < V_2$）denotes the probability of MPEG-2 residual MB when E is between V_1 and V_2. The function $Prob_2$（$V_1 \leqslant E < V_2$）denotes the probability of the optimal block size mode of the MB, which has the same spatial position in the frame as the MPEG-2 residual MB, is 16x16 when E is between V_1 and V_2. When $V_1=0$, 100, 200, …… 3900, set $V_2=V_1+100$. When $V_1=4000$, set $V_2=+\infty$. Figure 2 depicts the average value of functions $Prob_1$ and $Prob_2$ for the ten training sequence which is listed in Table 1.

From the Figure 2, we have the following observations: The smaller of the energy of residual MB, the higher of the probability of the optimal block size mode of the corresponding MB is 16x16 mode. For a MB, which contains more than one object and these objects may not move in the same direction, using only one MV may cause that only part of the MB have good motion compensation and the overall resulting residual energy can be large or the distribution of DCT coefficients may be unbalanced due to the mismatch in the remaining part of the MB.

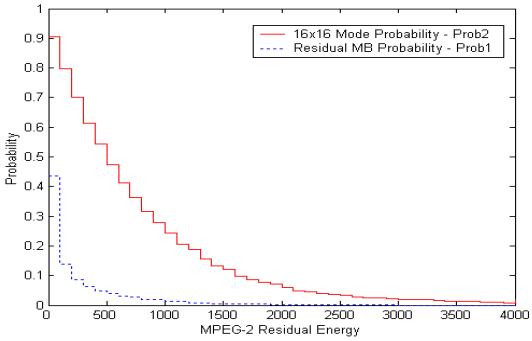


Fig 2. Statistics of 16x16 mode and MB

We also investigate the statistical correlation between the minimum search window size (MSWS) and the energy of the MPEG-2 residual block, which is showed in Table 2.

Residual Energy	MSWS			
	16x16	16x8	8x16	8x8
0~100	2	6	4	2
100~200	2	3	3	5
200~300	2	3	3	8
300~400	2	5	4	9
400~500	2	5	5	10
500~600	2	5	7	13
600~700	2	7	9	13
700~800	3	7	11	14
800~900	4	9	11	13
900~1000	9	11	11	14
1000~1100	10	16	13	15
1100~1200	14	12	13	14
1200~1300	9	13	12	16
1300~∞	16	14	13	14

Table 2. The statistics value of MSWS

The definition of MSWS is that it denotes the minimum search window size within which more than 80% MBs can find their optimal MVs. For example, in Table 2, when the optimal mode of MB is 8x8 mode and its corresponding MPEG-2 residual MB energy is between 300 and 400, the value of corresponding MSWS is 9. That is to say that among all the MBs whose MPEG-2 residual MB energy is between 300 and 400, there are at least 80% those MBs can find their optimal MVs in（2×9+1）×（2×9+1）search window. From the Table 2, we can see that the larger

of the MPEG-2 residual energy, the larger of the MSWS according the same block size mode. Here, all the MB optimal block size modes used to evaluate functions $Prob_1$ and $Prob_2$ and MSWS value are obtained in advance by using exhaustively full search algorithm in the MPEG-2 to H.264 transcoder using CPDT architecture.

3. Proposed algorithm

The above observations encourage the formation of our method: If the energy of MPEG-2 residual MB is very small, we can turn off the matching process from other block size modes since the performance of 16x16 mode is "good enough" and other modes require more MVs. On the other hand, if the energy of MPEG-2 residual MB is very large, we should directly split the MB into four 8x8 blocks to achieve better performance. If the early termination is not successful, we use the distribution of energy of four 8x8 block of a MPEG-2 residual MB to determine the final block size mode. In the end, the ME is only performed within in MSWS.

The detail of algorithm is as follows:

1. Initialization

Define several variables as:

LowT: threshold for 16x16 block type.

UpperT: threshold for 8x8 block type.

SumEng: accumulated energy of inter-predicted MPEG-2 residual MB.

N: accumulated number of MB used inter-code in MPEG-2 bit-stream.

Set LowT＝300, UpperT＝1000 and SumEng＝0. Visit each MB whose corresponding MPEG-2 MB is inter-coded and update the variable N by:

$$N = N + 1$$

Then perform the following steps:

2. Early termination

Calculate the energy of corresponding MPEG-2 residual MB E_{16x16} as defined above.

If E_{16x16}＜LowT, choose 16x16 as final block type. Else if E_{16x16}＞UpperT, choose 8x8 as final block type. Else go to the next step.

3. Block segmentation

Divide a 16x16 MB into four 8x8 blocks as shown in Figure 3. For each 8x8 block, use the variable array $E_{8x8}[i]$, i ＝ 0, 1, 2, 3 to represent their energy respectively.

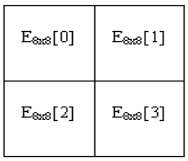


Fig 3. Division of 16x16 MB

Define several variables as:

MinBlock, MaxBlock: the minimum and maximum block energy among the four nonzero 8x8 block in the residual MB respectively.

AvgBlock: the average energy of the nonzero 8x8 blocks.

K: an empiric constant. In our experiments, it is set to 0.8

TopMB: the top part 8x8 block energy of MB, equal to $E_{8x8}[0] + E_{8x8}[1]$.

LowMB: the low part 8x8 block energy of MB, equal to $E_{8x8}[2] + E_{8x8}[3]$.

LeftMB: the left part 8x8 block energy of MB, equal to $E_{8x8}[0] + E_{8x8}[2]$.

RightMB: the right part 8x8 block energy of MB, equal to $E_{8x8}[1] + E_{8x8}[3]$.

Based on the number of $E_{8x8}[i]$＝0, there are three cases needed to be considered:

Case 1: Only one 8x8 block energy equal to zero.

Choose the 16x16 mode as the final mode. Because we think the overall prediction result is "good enough".

Case 2: Only two or three 8x8 block energy equal to zero.

Detect if only part of the MB is good motion compensated:

If Abs（MaxBlock－MinBloc）≤AvgBlock ×K, choose 16x16 as final block type. Else choose 8x8 as final block type.

Case 3: Non 8x8 block energy equal to zero.

Detect the distribution of residual energy:

If Abs（TopMB－LowMB）≥2×AvgBlock ×K, choose the 16x8 mode as the final mode, Else if Abs（LeftMB－RightMB）≥2×AvgBlock×K, choose 8x16 mode as the final mode, Else if Abs（MaxBlock－MinBlock）≤AvgBlock×K, choose 16x16 as the final block type, Else choose 8x8 as the final block type.

4. Threshold update

Threshold value should be adaptive with different scenes or sequences. In this paper, we update the two threshold: LowT and UpperT by following equations.

SumEng＝ SumEng ＋ E_{16x16}

LowT＝Min（（SumEng×1.5）／N, 300）

pperT＝Max（（SumEng×3）／N, 1000）

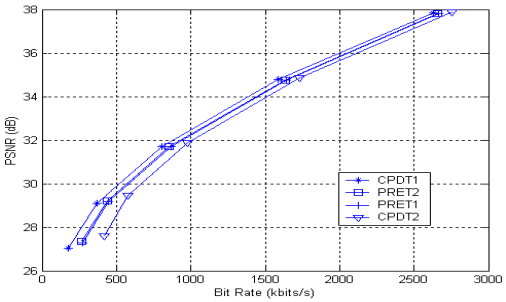
The value 1.5 and 3 is empiric constants, and the const 300 and 1000 are two residual energy threshold, which are determined from our extensive simulation.

5. Adaptive choosing search window size

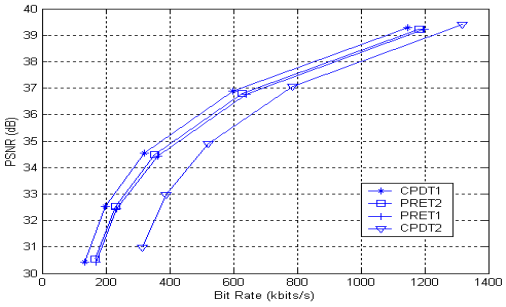
Based on our selected MB block type, associated MPEG-2 residual energy and the statistics value of MSWS listed in Table 2, we perform the ME only in the MSWS instead of doing it in the predefined search window size.

4. Simulation results

All of the transcoder is implemented using the H.264 JM 8.2 video codec [4] with search range of 16, 5 reference frames and rate-distortion optimization (RDO) disabled. The MPEG-2 encoder and the H.264 encoder both use only the frame prediction and frame picture and the "IPPPP…" sequence (include only I and P frames, no B frames) is used with a GOP size of 15 frames. The experiments are conducted on a computer based on an AMD Duron 1.1 GHz CPU with 256 MB RAM and Windows 2000 professional operating system. In experiments, the 90 frames of the two sequences (coastguard, foreman) are first encoded with MPEG-2 at 1.5 Mbps and at a frame rate of 30 fps, then transcoded to H.264 with quantizer values of 24, 28, 32, 36, and 40. PSNR, bit-rates, and computational complexity are analyzed for these two video sequences.



(a) Coastguard sequence



(b) Foreman sequence

Fig 4. Rate-distortion (PSNR vs. bit-rate) plots of four types transcoder for (a) Coastguard and (b) Foreman

Fig 4 shows the rate-PSNR curves of various type transcoders: two CPDT architecture transcoders, one use the full search for four block-sizes ($CPDT_1$) and another use only the 8x8 block-size for ME ($CPDT_2$) Because when only the 8x8 is enabled, the performance is acceptable, while when only the other single mode is enabled, the performance degradation is much more significant [5], the transcoder employing both of the fast mode selection method and the adaptive search window size choosing approach ($PRET_1$), and the partial re-encoding transcoder employing only the fast mode selection method ($PRET_2$).

From the Fig 4, we can see that both of $PRET_1$ and $PRET_2$ achieve appreciable performance improvement over $CPDT_2$ and are very closed to $CPDT_1$. Furthermore, The two curves of $PRET_1$, $PRET_2$ are hardly distinguishable.

Fig 5 shows the average computation amounts of two video sequences from the four types transcoders. The relative computational complexity of $PRET_1$ is only about 9.3% of $CPDT_1$. Since the average of the adaptive determined search window size is much smaller than the predefined search window size, in our simulations it is 16, the computation cost required in $PRET_1$ can be drastically reduced when compared to $PRET_2$.

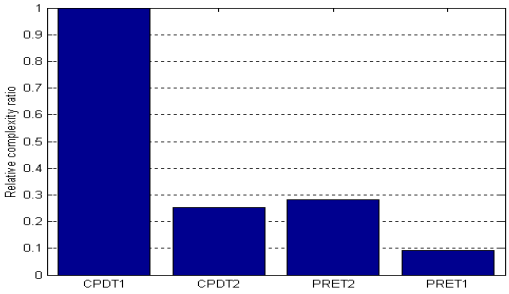


Fig 5. Average computation amount

5. Conclusion

In this paper, transcoding computational complexity is tremendously reduced by analysis and exploiting the MPEG-2 16x16 mode ME results. Simulation results demonstrate our proposed method can be used in developing real-time MPEG-2 to H.264 transcoder.

6. Acknowledgement

This work is supported by national nature science foundation of China under grant number 60302028.

7. 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)," in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, Mar.2003

[2] Hari Kalva. "Issues in H.264/MPEG-2 Video Transcoding". CCNC 2004. 5-8 Jan. 2004

[3] Jianning Zhang, Yuwen He, Shiqiang Yang, Yuzhuo Zhong, "Performance and complexity joint optimization for H.264 video coding" Circuits and systems, 2003. ISCAS '03.

[4] JVT Reference Software official version jm 8.2

[5] Yu-Kuang Tu, Jar-Ferr Yang, Yi-Nung Shen, Ming-Ting Sun. "Fast variable-size block motion estimation using merging procedure with an adaptive threshold". Multimedia and Expo, 2003. ICME'03.