Fast Block Size Selection for H.264 Video-Downsizing Transcoding

Zhuo-Yi Lu1,2, Wan-Chi Siu2 and Ke-Bin Jia1

1School of Electronic Information and Control Engineering, Beijing University of Technology, Beijing, China
2Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong

Abstract—This paper presents a fast variable block size selection method for inter macro-blocks in H.264 video downsizing transcoding. It makes use of the relationship between motion direction and edge direction of macro-blocks, and selects the optimal inter mode by measuring the dominant edge direction using the residual macro-block. Without fully decoding the compressed high-resolution video, Discrete Cosine Transform coefficients were extracted and reused in transform domain to construct the residual macro-block. Based on the measurement of dominant edge direction, only a small number of inter-modes are required for Rate Distortion Optimal evaluation. Experimental results show that the proposed method can reduce the computational complexity dramatically with little compression performance degradation.

I. INTRODUCTION

In recent years, downsizing video transcoding based on homogeneous compressed bit-stream, especially the transcoding from H.264 to H.264 has become emerging for various video applications. On observation, motion estimation in H.264 encoder is computationally intensive and typically requires at least 60% of the workload [1]. It is too time-consuming to directly cascade a H.264 decoder and encoder with brute-force mode decision for real-time applications. In order to reduce computational complexity, previous coding information such as motion vector and mode type have to be reused. Several methods have been presented in the literature focusing on mode decision. Reference [2] makes use of the characteristic of macro-blocks, such as macro-block’s edge intensity, and the difference between the current macro-block (MB) and that in the reference frame. Instead of using all 64 coefficients for each 8×8 block [3], [4] only uses the first three Alternating Current (AC) low-frequency coefficients to directly predict the block size partition from the set {16×16, 16×8, 8×16, 8×8} without full-search motion estimation for MPEG-2 to H.264 transcoding. Reference [5] provides an approach combining the fast intra mode decision and fast inter mode decision originated from [6]. These transcoding techniques based on homogeneous and heterogeneous bit-stream are useful for video downsizing transcoder in the pixel domain, but this conventional transcoding is computationally expensive caused by the fully decoding and re-encoding process. As a consequence, the DCT domain transcoding is introduced. Reference [7] implemented a DCT-based video downscaling transcoder using split and merge technique. However, the mismatch between the down-sampled DCT coefficients and the re-constructed motion vectors is still difficult to be overcome. This paper is focused on the mode decision methods of the transcoding from H.264 to H.264 videos.

The organization of this paper is as follows. In Section 2 we briefly review the inter mode decision in H.264. Our fast block size selection algorithm is described in Section 3. Simulation results are presented in Section 4. Finally, the concluding remarks are provided in the last section.

II. INTER MODE DECISION IN H.264

H.264 supports motion estimation using different block sizes. Each macro-block can be divided into big-partition sizes including 16×16, 16×8, 8×16 or P8×8. For each P8×8 partition, it can be further divided into small-partition sizes 8×8, 8×4, 4×8, 4×4 [8]. Mode decision can be made by comparing the Rate-Distortion (RD) cost of each inter mode and intra mode respectively. The mode with the minimal RD cost is selected as the best mode. The RD cost is calculated by (1).

\[
J(m) = D(m) + \lambda \cdot R(m),
\]

where \( D \) is the distortion; \( R \) is the number of coding bits associated with the chosen mode; \( \lambda \) represents the Lagrange multiplier for mode decision.

Note that each mode of the potential macro-block modes: \{SKIP, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4, I4MB and 116MB \} is evaluated by calculating its RD cost. This process consumes a large amount of computational effort in order to have good encoding efficiency. In the following section, we will introduce our proposed fast block size selection method used in down-sizing transcoding with low complexity and high quality.

III. FAST BLOCK SIZE SELECTION

Analysis of Macro-Block Residue and Movement Fast block size selection may be done based on the relationship between object movement direction and texture direction of a residual MB [9]. It is found that if the motion is uniform, the direction of the object movement is perpendicular to the dominant texture direction in the residual image [10]. The residual values vary in large difference due to the mismatch
of MB after motion estimation (ME), and certain edges in the residual MB.

The motion vectors (MV) obtained by ME represent the global motion direction of blocks, while the edge direction reflects the local motion activities within a block. Objects with different motion activities can be included in different sub-blocks by partitioning a MB along its edge direction. It is this reason why ME using variable block sizes can improve the compression performance. Based on this principle, we exploit the motion information extracted from the pre-coded residual MB to make inter mode decision.

During transcoding from MPEG-2 to H.264 [9], four neighbor 8×8 blocks are combined into one MB for H.264 re-encoder. In the H.264 re-encoder, the down-scaled 8×8 block is merged into large-partitions, or split into small-partitions. The DCT coefficients of each 8×8 block extracted from MPEG-2 bit-stream are used to calculate the edge parameters if it is decided to split an 8×8 block into small-partitions. In our scheme of transcoding from H.264 to H.264, we directly re-use the DCT coefficients of each 4×4 sub-blocks by partitioning a MB along its edge direction. It reflects the local motion activities within a block. Objects global motion direction of blocks, while the edge direction can be denoted by the edge direction in the MB movement and edge direction, it is observed that motion parameters can be illustrated in Fig. 2.

Since we have extracted the DCT coefficients of the original video, V and H can be directly calculated in the DCT domain. The inverse DCT for a 4×4 block is defined as:

$$p(k,l) = \sum_{n=-7}^{7} \sum_{m=-7}^{7} c(n,m) e^{2\pi j (n+1)k/8} e^{2\pi j (m+1)l/8}$$

$$C_{x} = \cos(\frac{\pi r}{2N}) = \cos(\frac{\pi r}{8}), \epsilon(r) = \begin{cases} 1 & r = 0 \\ 0 & \text{others} \end{cases}$$

In order to reduce the bit-rate and save computational time, we only use 3 out of the 16 DCT coefficients for each 4×4 block on the hypothesis that it has a limited impact on the overall performance. X(0,1), X(1,0) and X(1,1) are extracted to construct the residual MB, besides the DC coefficient X(0,0). Then the average intensity of a 4×4 block can be calculated as follows:

$$S_{v} = \frac{1}{16} \sum_{i=0}^{3} \sum_{j=0}^{3} [e(0) e(0) X(0,0) C_{x} C_{y} + e(0) e(0) X(1,0) C_{x} C_{y} + e(0) e(0) X(0,1) C_{x} C_{y} + e(0) e(0) X(1,1) C_{x} C_{y} + e(0) e(0) X(0,0) C_{x} C_{y} + e(0) e(0) X(1,0) C_{x} C_{y} + e(0) e(0) X(0,1) C_{x} C_{y} + e(0) e(0) X(1,1) C_{x} C_{y}]$$

The edge parameters for an 8×8 block can be formulated as (6), where QP is set to 28:

$$V = \frac{1}{16+28} [e(0) e(0) C_{v} C_{v} [X_{00}(0,0) - X_{00}(0,0) + X_{00}(0,0) - X_{00}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{01}(0,0) - X_{01}(0,0) + X_{01}(0,0) - X_{01}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{10}(0,0) - X_{10}(0,0) + X_{10}(0,0) - X_{10}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{11}(0,0) - X_{11}(0,0) + X_{11}(0,0) - X_{11}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{00}(0,0) - X_{00}(0,0) + X_{00}(0,0) - X_{00}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{01}(0,0) - X_{01}(0,0) + X_{01}(0,0) - X_{01}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{10}(0,0) - X_{10}(0,0) + X_{10}(0,0) - X_{10}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{11}(0,0) - X_{11}(0,0) + X_{11}(0,0) - X_{11}(0,0)], H = \frac{1}{16+28} [e(0) e(0) C_{v} C_{v} [X_{00}(0,0) - X_{00}(0,0) + X_{00}(0,0) - X_{00}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{01}(0,0) - X_{01}(0,0) + X_{01}(0,0) - X_{01}(0,0)] + e(0) e(0) C_{v} C_{v} [X_{10}(0,0) - X_{10}(0,0) + X_{10}(0,0) - X_{10}(0,0)]$$

$$+ e(0) e(0) C_{v} C_{v} [X_{11}(0,0) - X_{11}(0,0) + X_{11}(0,0) - X_{11}(0,0)].$$

**A. Edge Parameter Extraction**

For a 2:1 video down-size transcoding each MB in the high resolution video basically has to be encoded and re-encoded by H.264. During this process, the coding information of the original video can be extracted from the H.264 bit-stream.

According to the analysis of the relationship between MB movement and edge direction, it is observed that motion direction can be denoted by the edge direction in the residual MB. To improve the performance, the edge direction can be obtained by calculating residual MB in terms of DCT coefficients in the transform domain instead of the pixel-domain to avoid full-decoding. The residual MB is firstly divided into four 8×8 blocks, and each 8×8 block is further divided into four 4×4 blocks as shown in Fig. 1.

![Figure 1. An 8×8 block](image)

Let $S_{v0}, S_{v1}, S_{h0}$ and $S_{h1}$ represent the average intensity of each 4×4 block, and $p(k,l)$ be the intensity value of each pixel, for $k, l = 0, 1, ..., 7$. We have,

$$S_{v} = \frac{1}{16} \sum_{i=0}^{3} \sum_{j=0}^{3} [p(4u+i,4v+j), u, v = 0, 1].$$

We define two edge parameters, vertical edge parameter $V$ and horizontal edge parameter $H$.

$$V = \frac{(S_{v0} - S_{v1}) + (S_{v1} - S_{v0})}{S} H = \frac{(S_{h0} - S_{h1}) + (S_{h1} - S_{h0})}{S}$$

where the scaling factor $S$ equals the quantization step (QP) in the H.264 re-encoder. The physical meanings of the two parameters can be illustrated in Fig. 2. $H$ represents the intensity differences between the upper and lower parts in a block, and $V$ represents the intensity differences between the left and right parts.

**B. Fast Block Size Selection**

In down-sizing transcoding from H.264 to H.264, each 16×16 macro-block is down-sampled to 8×8 block, as it is shown in Fig. 3. A 16×16 MB in the downsized video consists of four neighbor macro-blocks (A, B, C, D) in the original high resolution video.

![Figure 3. Video down-sampling](image)

For big-partition prediction, the edge parameters $big\_V$ and $big\_H$ are defined as:

$$big\_V = (S_{VR} - S_{LR}) + (S_{CU} - S_{DL}),$$

$$big\_H = (S_{VR} - S_{LR}) + (S_{CU} - S_{DL}).$$

$$H$$

$$V$$

**Figure 2. Physical meanings of $H$ and $V$.**

Since we have extracted the DCT coefficients of the original video, $V$ and $H$ can be directly calculated in the DCT domain. The inverse DCT for a 4×4 block is defined as:
\[ S = \frac{1}{16 \times 28} \left[ (0)(0)(0)(0)C_4^x \left[ X_{00}(0,0) + X_{01}(0,0) + X_{10}(0,0) + X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^y \left[ X_{00}(0,0) + X_{01}(0,0) + X_{10}(0,0) + X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^z \left[ X_{00}(0,0) + X_{01}(0,0) + X_{10}(0,0) + X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^w \left[ X_{00}(0,0) + X_{01}(0,0) + X_{10}(0,0) + X_{11}(0,0) \right] \right] \]

For small-partition prediction, the vertical edge parameter \( small_v \) and horizontal edge parameter \( small_h \) are formulated as:

\[
small_v = \frac{1}{16 \times 28} \left[ (0)(0)(0)(0)C_4^x \left[ X_{00}(0,0) - X_{01}(0,0) + X_{10}(0,0) - X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^y \left[ X_{00}(0,0) - X_{01}(0,0) + X_{10}(0,0) - X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^z \left[ X_{00}(0,0) - X_{01}(0,0) + X_{10}(0,0) - X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^w \left[ X_{00}(0,0) - X_{01}(0,0) + X_{10}(0,0) - X_{11}(0,0) \right] \right],
\]

\[
small_h = \frac{1}{16 \times 28} \left[ (0)(0)(0)(0)C_4^x \left[ X_{00}(0,0) - X_{01}(0,0) + X_{10}(0,0) - X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^y \left[ X_{00}(0,0) - X_{01}(0,0) + X_{10}(0,0) - X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^z \left[ X_{00}(0,0) - X_{01}(0,0) + X_{10}(0,0) - X_{11}(0,0) \right] + (0)(0)(0)(0)C_4^w \left[ X_{00}(0,0) - X_{01}(0,0) + X_{10}(0,0) - X_{11}(0,0) \right] \right].
\]

(8)

According to \( H \) and \( V \) of both big-partition and small-partition, each 16×16 and 8×8 block can be classified into one of the four categories as tabulated in Table I respectively.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Edge direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>H</td>
</tr>
<tr>
<td>(</td>
<td>0</td>
</tr>
<tr>
<td>(</td>
<td>0</td>
</tr>
<tr>
<td>(</td>
<td>H</td>
</tr>
</tbody>
</table>

Generally speaking, the homogeneous areas in the video sequence such as the background are always coded using 16×16 mode or SKIP mode. On the contrary, a small partition is chosen for MB with complex motion or texture. From our observation, in fact, the probability of using SKIP and 16×16 modes is larger than other modes for common video sequences. To make the block size selection more reliable for practical application, a few possible modes are added to the prediction list by sacrificing little computational complexity. For example, when classifying the big-partition mode for a 16×16 block, we just eliminate 8×16 mode if it has horizontal dominant direction edge.

The detailed block size selection method we proposed is carried out in two steps. The flowchart is illustrated in Fig. 4. Firstly, edge parameters \( V \) and \( H \) are extracted during H.264 decoding process, and kept in a data file. Then, by comparing the value of vertical and horizontal edge parameters, big-partition and small-partition prediction are operated as following:

Firstly, we classify each 16×16 block as modes 16×16, 8×16, 8×8 or P8×8 making use of the edge directions. This is done by measuring its big-partition edge parameters:

- 8×8 is eliminated for horizontal dominant direction edge;
- 16×8 is eliminated for vertical dominant direction edge;
- 16×8 and 8×16 are eliminated for diagonal direction edge;
- Otherwise check SKIP, 16×16, 16×8, 8×16, P8×8 for no obvious dominant edge.

Secondly, classify each P8×8 block as modes 8×8, 8×4, 4×8 or 4×4 making use of the edge direction. This is done by measuring its small-partition edge parameters:

- 4×8 is eliminated for horizontal dominant direction edge;
- 8×4 is eliminated for vertical dominant direction edge;
- 8×4 and 4×8 are eliminated for diagonal direction edge;
- Otherwise check 8×8, 8×4, 4×8, and 4×4 for no obvious dominant edge.

III. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed algorithm was implemented using the H.264 reference software JM12.2. In addition, merge and split mode decision method in reference [9] was also implemented for comparison. In our simulations, four sequences in CIF format (352×288) are downsized to QCIF format (176×144) with the down-sizing ratio 2:1. The first frame was encoded as intra-frame (I-frame), and the remaining frames were encoded as inter-frame (P-frame).
used to calculate the dominant edge direction, which allows to construct residual MB in the transform domain. More candidate modes are included in the RDO process to guarantee the performance of transcoding. Experimental results show that our proposed fast block size selection method dramatically reduces computational complexity compared with the full re-encoding transcoding while maintains a similar rate-distortion performance.

Further work is now being done on obtaining the residual DCT coefficients without fully decoding the compressed bit-stream. It is obvious that a direct use of DCT coefficients in the DCT domain can achieve low computational complexity. It is good that if we can convert the transform coefficients from the original transform domain to the new transform domain directly [11, 12], which is a difficult task. At the present moment we can only partially do this, and we expect new results on this respect for variable block sizes can be available soon.

### REFERENCES


