Fast Decision of Block size, Prediction Mode and Intra Block for H.264 Intra Prediction

Guided by – Dr. K. R. Rao
Gaurav Hansda
UTA ID – 1000721849
gaurav.hansda@mavs.uta.edu

OBJECTIVE
The objective is to implement a variance-based algorithm for block size decision, an improved filter-based algorithm for prediction mode decision, and a selection algorithm for intra block decision that exploits the relation between the rate-distortion characteristic and the best coding type. All algorithms are to be implemented using JM reference software [17]. This is for the latest video coding standard for H.264 [19].

MOTIVATION
The H.264/Advanced video coding (AVC) standard [1] achieves better performance than the previous video coding standards and has become a key technology for multimedia communications and consumer electronics. However, like the other advanced coding techniques adopted earlier, it also brings in significant computational overhead to the encoder. The mode decision of H.264 is more complicated and time-consuming than those of previous coding standards. In fact, it is the most computationally intensive component of an H.264 encoder, more so for the High Profile. Various profiles of H.264 are shown in Fig.1. Therefore, fast algorithms for mode decision are needed.

The high performance of this standard is achieved by the adoption of advanced coding tools, such as spatial-domain adaptive intra directional prediction, variable block-size motion estimation (Fig. 2), multiple reference frames, and rate-distortion (R-D) optimization [2], [3]. In particular, the H.264 High Profile is further developed for applications such as professional film production, digital video broadcasting, and high-definition TV and video disc. One notable difference of this new profile is that it adopts the 8×8 intra prediction as a new coding tool to improve the R-D performance.

Fig. 3 shows a block diagram of the video coding layer (VCL) for a macroblock [2]. The input video signal is split into macroblocks, the association of macroblocks to slice groups and slices are selected, and then each macroblock of each slice is processed as shown. An efficient parallel processing of macroblocks is possible when there are various slices in the picture.
Fig. 1. Profiles in H.264 [17]

Fig. 2. Segmentations of the macroblock for motion compensation [2]
As shown in Fig. 4, the mode decision process of an H.264 compliant encoder entails a three-stage hierarchy of operations: inter/intra block decision, block size decision, and prediction mode decision [4]. There are two important implications of this hierarchical structure of mode decision. First, to ensure the correctness of the decision at upper layer and second, to ensure early termination is executed accurately and as early as possible.

Most fast mode decision algorithms developed so far, only deal with a single stage of the mode decision hierarchy [5]-[15] and fail to achieve the best possible complexity reduction. In this project more focus is given to left branch of the mode decision hierarchy namely, inter/intra block decision of inter frames, block size decision of intra blocks, and the prediction mode decision of intra blocks.

The techniques employed in the proposed algorithms are inspired by several factual observations. For block size decision, it is known that the block size of the best coding mode is highly correlated with texture complexity [16]. In addition, the variance of a macroblock (MB) in the spatial domain, equivalent to the energy of AC components in the discrete cosine transform domain, is a low-cost and effective measurement of texture complexity [16].
For prediction mode decision, most filter-based algorithms use edge detectors to determine dominant edges and limit the search of the best prediction mode to those along the detected edges [5]-[12]. However, the information between neighboring blocks is not taken into account. An improved traditional filter-based prediction mode decision algorithm can be obtained by incorporating contextual information [4]. The resulting algorithm works effectively for the High Profile of H.264. When using the Intra4x4 mode, each 4x4 block is predicted from spatially neighboring samples as illustrated in Fig. 4. Eight prediction modes are shown in Fig. 5. Fig. 6 shows a 4x4 luma block that is to be predicted. For the predicted samples [a,b, ... ,p] for the current block, the above and left previously reconstructed samples [A,B, ... ,M] are used according to direction modes. The arrows in Fig. 6 indicate the direction of prediction in each mode. For mode 0 (vertical) and mode 1 (horizontal), the predicted samples are formed by extrapolation from upper samples [A, B, C, D] and from left samples [I, J, K, L], respectively.
For mode 2 (DC), all of the predicted samples are formed by mean of upper and left samples [A, B, C, D, I, J, K, L]. For mode 3 (diagonal down left), mode 4 (diagonal down right), mode 5 (vertical right), mode 6 (horizontal down), mode 7 (vertical left), and mode 8 (horizontal up), the predicted samples are formed from a weighted average of the prediction samples A–M. For example, samples ‘a’ and ‘d’ are, respectively, predicted by round \((I/4 + M/2 + A/4)\) and round \((B/4 + C/2 + D/4)\) in mode 4, also by round \((I/2 + J/2)\) and round \((J/4 + K/2 + L/4)\) in mode 8. The encoder may select the prediction mode for each block that minimizes the residual between the block to be encoded and its prediction [17].

For prediction of each 8x8 luma block, one mode is selected from the 9 modes, similar to the (4x4) intra-block prediction [17]. For prediction of all 16x16 luma components of a macroblock, four modes are available as shown in Fig. 7. For mode 0 (vertical), mode 1 (horizontal), mode 2 (DC), the predictions are similar with the cases of 4x4 luma block. For mode 4 (plane), a linear ‘plane’ function is fitted to the upper and left-hand samples H and V. This works well in areas of smoothly-varying luminance [19].

For intra/inter block decision, it is observed that the adoption of fast motion estimation greatly reduces the computational complexity of inter block coding. As a result, the time required for checking the modes of intra blocks becomes relatively significant compared to that of inter blocks. In addition, although the number of intra blocks of an inter frame is usually small, it takes a considerable amount of computation to come to the decision exhaustively [4]. Therefore, a fast algorithm for intra block decision is needed.

![Fig. 6. Intra 4x4 prediction mode directions (vertical, 0; horizontal, 1; DC, 2; diagonal down left, 3; diagonal down right, 4; vertical right, 5; horizontal down, 6; vertical left, 7; and horizontal up, 8) [17]](image-url)
Proposed Algorithm for Block Size Decision

Block size is highly correlated with texture complexity. Variance of block corresponds to the total energy of the AC coefficients of the block; hence it is a good measurement of the texture complexity. Thus variance based classification of texture complexity [16] is used in the algorithm. Variance for 16x16 macroblock is given as:

\[
VARIANCE = \sum_{i=0}^{15} \sum_{j=0}^{15} [Y(i,j)]^2 - \frac{1}{256} \left[ \sum_{i=0}^{15} \sum_{j=0}^{15} Y(i,j) \right]^2 \tag{1}
\]

where \(Y(i, j)\) is the luminance value of the pixel at \((i, j)\). The texture complexity of an MB is classified according to its variance as follows:

\[
\text{complexity} = \begin{cases} 
\text{high, if variance} > T_1 \\
\text{low, otherwise}
\end{cases}
\tag{2}
\]

where \(T_1\) is a threshold, which is set to 92 735 for 8 bit PCM. The determination of this value is found in [16].

The flow of the proposed block size decision is shown in Fig. 8. If the calculated variance is above the threshold, Intra4MB and Intra8MB modes are selected; otherwise, Intra8MB and Intra16MB are chosen. This is a simple way to skip the examination of Intra4MB mode.
Improved Prediction Mode Decision

This algorithm is based on the filter-based algorithm described in [8]. However this algorithm only considers the edge information of the current block and does not take correlation between blocks into account. Hence use of Most Probable Mode (MPM) is incorporated to the algorithm described in [8].

The MPM, which takes advantage of the spatial correlation of the prediction modes between the neighboring blocks and the current block for coding, is defined as the prediction mode of the left or the upper neighbor, whichever has the smaller prediction mode number. If either of these two neighbors is unavailable, a default DC mode is considered. Since only one bit is needed to signal the MPM, the other eight modes can be effectively coded with three bits [1], keeping the total number of bits required for representing the prediction mode of the current block to four.

The block diagram of the improved algorithm for prediction mode decision is shown in Fig. 9. For 4×4 luma block, three intra prediction modes along the direction of the detected edge and its adjacent directions are chosen as candidate modes. Since the MPM is more likely than the DC mode to be the best mode, always include the MPM as the candidate mode. If the MPM is the same as any of the other three candidate modes, replace it by the DC mode.
Selective Intra Block Decision

Intra block decision, which is performed for inter frames, occupies a considerable percentage of the total computations of inter-frame coding. Intra16MB takes much less computation time than the other modes. In addition, it has been found that the rate-distortion (R-D) cost of Intra16MB is close to that of Intra4MB [13]. Therefore, it is less probable for Intra4MB to be the best mode if the R-D cost of Intra16MB is much larger than that of the best inter MB mode. Hence scaled R-D cost is used.

The R-D cost, denoted by $J_{mode}$, of an MB mode is defined as follows:

$$J_{mode}(mode, QP, \lambda) = SSD(mode) + \lambda R(mode, QP)$$  \hspace{1cm} (3)

where $mode \in \{\text{Inter modes, Intra4MB, Intra8MB, Intra16MB}\}$, QP is the quantization parameter, $\lambda$ is a Lagrange multiplier, $SSD(\cdot)$ denotes the sum of squared difference, and $R(\cdot)$ denotes the total number of bits used for coding the selected mode, the macroblock header, and the residual information [4].
The scaled R-D cost is defined as [4]

\[
\hat{j}_{\text{mode}} = \frac{j_{\text{mode}}}{\lambda}
\]  \hfill (4)

where \(\lambda\) is a scaling factor. The advantage of using the scaled R-D cost is that it lies in the same range under different QPs, which makes the analysis simpler and, most importantly, a universal threshold can be used. An MB is very likely to be inter-coded when the R-D cost of the best inter mode is much lower than that of the Intra16MB. On the other hand, the intra modes would be selected as the best mode when the R-D cost of the best inter mode is higher than or close to the R-D cost of Intra16MB. Therefore, it is obvious to predict that an MB is less probable to be intra coded if the R-D cost difference is small. Denoting the scaled R-D cost differences between Intra16MB and the inter MB mode by \(d\hat{J}\), based on the above observation, both Intra4MB and Intra8MB are get skipped if \(d\hat{J}\) is small. This prediction rule has a significant computational advantage for inter-frame coding since the examination of Intra4MB and Intra8MB is computationally more complex.

The proposed intra block decision algorithm consists of the following steps [4].

1) For each MB in an inter frame, examine all inter MB modes and store the R-D cost of the best inter MB mode.
2) Examine the Intra16MB mode and store its R-D cost.
3) If \(d\hat{J} > T_2\), check Intra4MB and Intra8MB.
4) Select the MB mode that has the minimum R-D cost among all visited modes as the best mode.

The threshold \(T_2\) in Step 3 is set to \(-150\), which is empirically determined according to the histograms of the R-D cost difference \((\hat{j}_B - \hat{j}_{\text{Intra16MB}})\) where \(\hat{j}_B\) is the R-D cost of the best inter mode. The same \(T_2\) is applied to all the sequences used. It should be noted that, since \(\lambda\) in (4) is a function of QP [1] and controls \(d\hat{J}\), the algorithm is adaptive to QP. No manual tuning is required.

**RESULTS**

The JM reference software version 17.2 is implemented [18]. The conditions of the experiments are as follows.

1) Run on a PC with Intel Core i3 2.27GHz processor and 3.00 GB RAM.
2) Set the QP value to 16, 20, 24, and 28.
3) Enable the R-D optimization.
4) Choose CABAC as the entropy coding method.

*Hall.qcif* (Fig. 10) is used for this experiment. The performance of JM reference software is observed with respect to varying values of quantization parameter (QP). Table 1 shows the variation of peak signal
to noise ratio (PSNR), bit rate and total encoding time with QP values set at 28, 24, 20 and 16. Fig. 11 shows the respective variation in graphical format.

Table I
Performance of hall (QCIF) in JM reference software version 17.2

<table>
<thead>
<tr>
<th>QP</th>
<th>PSNR(dB)</th>
<th>Bit rate (kbits/s)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>38.685</td>
<td>697.64</td>
<td>20.022</td>
</tr>
<tr>
<td>24</td>
<td>41.378</td>
<td>961.59</td>
<td>30.875</td>
</tr>
<tr>
<td>20</td>
<td>44.118</td>
<td>1359.14</td>
<td>34.094</td>
</tr>
<tr>
<td>16</td>
<td>47.085</td>
<td>1930.5</td>
<td>37.767</td>
</tr>
</tbody>
</table>

Fig. 11. Performance curve for Hall (QCIF) (a)Bit rate vs PSNR. (b)Encoding time vs PSNR.
REFERENCES


[18] JM software – http://iphome.hhi.de/suehring/tml/ This software is a product of Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. The latest version of JM Software is 17.2