Qualitative DC Mode Conditions for Fast Intra Prediction Mode Decision in H.264/AVC

Do Quan and Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Republic of Korea
E-mails: {viequando,hoyo}@gist.ac.kr

Abstract—In this paper, we present a simple but efficient algorithm for intra prediction mode decision in H.264/AVC. Based on our investigation, the DC mode appears to be the superior prediction mode among various candidates. We propose an intra-mode decision algorithm where the DC mode is chosen as a candidate for the best prediction mode. Our experimental results show that the proposed algorithm saves 80.82% of the encoding time, compared to the H.264 reference software on average, while reducing the negligible peak signal-to-noise ratio (PSNR) values and slightly increasing the bit rate. In order to reduce the bitrate increment of the above-mentioned algorithm and refine our DC mode condition, we propose another fast intra prediction mode decision algorithm where we use a condition of block boundary to select the DC mode. Our experimental results show that this proposed algorithm not only increases PSNR values and decreases bitrates, but also saves the encoding time.

Keywords-H.264/AVC, Video coding, Intra prediction, Mode decision.

I. INTRODUCTION

H.264 [1] is a powerful video coding standard in terms of both PSNR value and visual quality. One of the coding approaches uses variable block sizes of macroblock (MB) modes. Although H.264 provides a rate-distortion optimization (RDO) technique for determining the best mode, it still consumes much time and remains complex due to heavy computational load in order to check all combinations of modes for each MB. These are the obstacles to implementation of an encoder for real-time applications, such as video telephony and video conferencing.

Figure 1 summarizes the mode decision algorithm based on Lagrangian cost function which is meant to minimize mode cost in H.264 [2]. For intra prediction, H.264 uses three different block sizes: intra-16×16, intra-8×8, and intra-4×4 [3]. Intra-4×4 has nine different modes, such as vertical mode, horizontal mode, DC mode, and other six diagonal modes formed by combinations of angles for intra-luma prediction. If high profile is chosen, we have intra-8×8 block size. Intra-8×8 using 8×8 transform consists of nine modes which are similar to intra-4×4 modes. Intra-16×16 has four modes including the vertical mode, horizontal mode, DC mode, and plane mode. For intra-chroma prediction, intra-8×8 has four modes but these modes have different orders compared to intra-16×16 mode orders.

In order to find the best mode for one macroblock, H.264 uses RDO technique for 592 modes: 9 intra-4×4 modes, 4 intra-16×16 modes for luma prediction, and 4 intra-8×8 modes for chroma prediction if high profile is not chosen. Otherwise, using the high profile, H.264 uses the RDO technique for 736 modes since it additionally calculates 9 intra-8×8 modes for luma prediction. Those numbers of modes are so huge for one macroblock and consume much time for sum square different calculation in the RDO technique. In order to solve this problem, we propose two algorithms in order to reduce the number of modes for intra prediction mode decision.

In terms of the studies on H.264, there have been many proposed algorithms for fast intra prediction mode decision. In these algorithms, researchers have tried to reduce complexity while ensuring acceptable visual quality. Tsai et al. [4] used an intensity gradient technique to make an efficient intra prediction. In addition, Pan et al. [5] proposed a fast mode decision algorithm for intra-frame coding using local edge information of each macroblock. La et al. [6] used the dominant edge direction to make fast mode decision. Lin et al. [7] developed a simple algorithm based on direction detection of the edge inside the block. Furthermore, Wang et al. [8] proposed a fast algorithm based on the dominant edge strength. These algorithms have limitations in coding performances and

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-C1099-1011-0003)

978-1-4244-8072-2/10/$26.00 ©2010 IEEE
time consumption; therefore, in this paper, we propose our algorithms to improve these performance metrics.

In this paper, we first propose a simple but efficient algorithm for intra prediction mode decision. Instead of checking all combinations of modes for each MB, we directly check the best mode which is observed from our research. We expect to get a significantly reduced coding time, an acceptable increase in bitrates and a decrease in PSNR values from our experiments. However, in order to reduce the increment of the bitrate of the first proposed algorithm, we propose another fast intra prediction mode decision algorithm using a condition of block boundary to select the DC mode. Finally, with the same test sequences and the same initial conditions, we compare the results of our two proposed algorithms with those of the above-mentioned algorithms.

II. PROPOSED ALGORITHMS

In this section, we represent two proposed algorithms for fast intra prediction mode decision. First, we propose an unconditional DC mode decision algorithm. We also propose another algorithm using a condition of block boundary to select the DC mode.

A. Fast Intra Prediction Mode Decision Algorithm Using the Unconditional DC Mode

In this subsection, we propose a fast intra prediction mode decision algorithm by using a characteristic of the DC mode in intra-4×4 and intra-16×16 for intra-luma prediction. We will present how to determine the best mode for both intra-chroma and intra-luma prediction.

First of all, we determine the best mode for intra-chroma prediction. In the proposed algorithm, four intra-8×8 modes for intra-chroma prediction are calculated by the sum of absolute Hadamard transform differences (SATD). The best mode has the minimum SATD.

For luma-prediction, we use fidelity range extensions (FRExt) as a setting option for this algorithm. Through our research on intra-4×4 and intra-16×16, we investigate that the DC mode is the superior prediction mode among the various candidates. An example of this investigation for "Tempe" sequence was shown in [9]. Other test sequences, i.e., "Mobile" sequence as shown in Figure 2, have similar statistics of mode selection [10]. Therefore, the DC mode is chosen as the best candidate for the prediction mode. This choice is strongly agreed for 4×4 intra prediction mode in [11], and both 16×16 luma prediction and 8×8 intra-chroma prediction in [12]. In this way, we select the DC mode as the best prediction mode for intra-4×4 and intra-16×16 predictions. We repeat the above-mentioned processes for the next macroblock.

In this way, the proposed algorithm significantly reduces the numbers of modes instead of checking 592 or 736 modes as H.264 uses in the RDO technique. Hence, the encoding time will be reduced. This will be confirmed through our experimental results in Section III.

Figure 3 shows the flowchart of the proposed intra-mode decision algorithm using the unconditional DC mode, which is referred to as algorithm 1. The proposed algorithm consists of

![Flowchart of fast intra-mode decision algorithm using the unconditional DC mode (Algorithm 1).](image)

the following steps.

Step 1: Determine the best mode among the four intra-8×8 modes for intra-chroma prediction.

Step 2: Select the DC mode as the best mode for intra-4×4 luma prediction.

Step 3: Select the DC mode as the best mode for intra-16×16 luma prediction.

Step 4: Repeat the above processes for the next macroblock until we check the last of macroblocks.
In order to reduce the increment of the bitrate of the proposed algorithm 1, we propose another fast intra prediction mode decision algorithm using a condition of block boundary to select the DC mode. Using the same test sequences and the same initial conditions, we compare the results of our two proposed algorithms with those of the existing algorithms. Through this comparison, we can see the improvement by using the refinement of the DC mode condition.

B. Fast Intra Prediction Mode Decision Algorithm Using a Condition of Block Boundary to Select the DC Mode

This subsection presents our proposed algorithm for fast intra prediction mode decision by checking boundary pixels of every block to select the DC mode.

The main motivation for the proposed algorithm is represented in Figure 4. If the pixel values of block boundary are the same or similar values, we conclude that all values inside the block are the same or similar to the pixel values of block boundary. This means that we can use the DC mode as the best mode for fast intra prediction mode decision.

![Figure 4. Condition to select the DC mode.](image)

For intra-chroma prediction, we use the RDO technique given in H.264 to determine the best mode. This means that four intra-8×8 modes for intra-chroma prediction are calculated by SATD and the best mode has the minimum SATD.

For intra-luma prediction, we apply the following procedure in both intra-4×4 and intra-16×16. First, we determine boundary variance and threshold for this block. Then, we compare the variance with the threshold to determine the best mode. If the variance is smaller than the threshold, this means that the pixel values on boundary are the same or similar; in this case, we choose the best mode as the DC mode. Otherwise, we use the RDO technique given in H.264 among other modes to determine the best mode for this block. The threshold is adaptive since it is determined by a quantization step size for each block.

From above analysis, this proposed algorithm uses more time than our proposed algorithm 1. However, it also reduces the numbers of modes instead of checking 592 or 736 modes as H.264 uses in the RDO technique. Hence, the entire encoding time of our program will be reduced. This will also be confirmed through our experimental results in Section III. Finally, we summarize this proposed algorithm, which is referred to as algorithm 2, as follows.

Algorithm 2: Fast intra prediction mode decision algorithm using a condition of block boundary to select the DC mode

1: // Intra-chroma prediction
2: best mode ← best mode by RDO for 8×8 intra chroma.
3:
4: // Intra-luma prediction
5: // Determine the best mode for intra-4×4
6: \( \sigma_{b1} = \text{boundary variance of intra-4×4} \)
7: \( th_1 = (Q_{\text{step}}^2 + 8)/16 \)
8: if \( \sigma_{b1} < th_1 \) then
9: best mode ← DC mode
10: else
11: best mode ← best mode by RDO
12: end if
13:
14: // Determine the best mode for intra-16×16
15: \( \sigma_{b2} = \text{boundary variance of intra-16×16} \)
16: \( th_2 = (Q_{\text{step}}^2 + 128)/256 \)
17: if \( \sigma_{b2} < th_2 \) then
18: best mode ← DC mode
19: else
20: best mode ← best mode by RDO
21: end if

III. EXPERIMENTAL RESULTS AND DISCUSSION

In Section II, we propose two algorithms: a fast intra decision algorithm using the unconditional DC mode and a fast intra decision algorithm using a condition of block boundary to select the DC mode. We briefly call the two proposed algorithms in the above order as proposed algorithms 1 and 2.

A. Coding Conditions

We implement the two proposed algorithms on the reference software JM11.0 [13]. In addition, we examine various CIF and QCIF sequences, which are adopted as the test sequences in MPEG standard, i.e., "Coastguard", "Silent", "Grandma", "Mobile" for QCIF sequences, and "Bridge close", "Football", "Mobile", "Silent" for CIF sequences. The system platform is Intel Pentium(R) 4, a processor of speed 3.6GHz, 2.00 GB of DDR RAM, and Microsoft Windows XP for the extensive simulations. Table I shows the simulation conditions in detail [14].

<table>
<thead>
<tr>
<th>TABLE I. ENCODING PARAMETERS FOR FAST INTRA-MODE DECISION ALGORITHM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>RDO mode</td>
</tr>
<tr>
<td>GOP structure</td>
</tr>
<tr>
<td>Hadamard transform</td>
</tr>
<tr>
<td>Quantization parameters</td>
</tr>
<tr>
<td>Number of frames</td>
</tr>
</tbody>
</table>

We compare the two proposed algorithms with JM11.0 in terms of PSNR difference, percentage bitrate difference, and
<table>
<thead>
<tr>
<th>Test sequence</th>
<th>Proposed algorithm 1</th>
<th>Proposed algorithm 2</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ΔPSNRY [dB]</td>
<td>ΔBitrate [%]</td>
<td>ΔTime [%]</td>
<td>ΔPSNRY [dB]</td>
<td>ΔBitrate [%]</td>
</tr>
<tr>
<td>Coastguard (QCIF)</td>
<td>-0.05</td>
<td>+5</td>
<td>-85.06</td>
<td>+0.03</td>
<td>+4.78</td>
</tr>
<tr>
<td>Silent (QCIF)</td>
<td>-0.07</td>
<td>+4.86</td>
<td>-80.92</td>
<td>+0.05</td>
<td>+3.81</td>
</tr>
<tr>
<td>Grandma (QCIF)</td>
<td>-0.15</td>
<td>+3.76</td>
<td>-80.63</td>
<td>+0.08</td>
<td>+3.15</td>
</tr>
<tr>
<td>Mobile (QCIF)</td>
<td>-0.36</td>
<td>+3.33</td>
<td>-78.66</td>
<td>+0.23</td>
<td>+1.55</td>
</tr>
<tr>
<td>Bridge close (CIF)</td>
<td>-0.17</td>
<td>+5.79</td>
<td>-77.41</td>
<td>+0.1</td>
<td>+3.77</td>
</tr>
<tr>
<td>Football (CIF)</td>
<td>-0.02</td>
<td>+4.83</td>
<td>-81.65</td>
<td>0</td>
<td>+3.07</td>
</tr>
<tr>
<td>Mobile (CIF)</td>
<td>-0.2</td>
<td>+5.21</td>
<td>-80.24</td>
<td>+0.21</td>
<td>+2.01</td>
</tr>
<tr>
<td>Silent (CIF)</td>
<td>-0.04</td>
<td>+5.95</td>
<td>-81.99</td>
<td>+0.02</td>
<td>+5.21</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>-0.13</td>
<td>+4.84</td>
<td>-80.82</td>
<td>+0.09</td>
<td>+3.42</td>
</tr>
</tbody>
</table>

(*) Proposed algorithms 1 and 2 denote fast intra prediction mode decision algorithms using the unconditional DC mode, and a condition of block boundary to select the DC mode, respectively.

Run-time percentage difference. These performance metrics are defined as follows:

\[
\Delta \text{PSNRY} = \text{PSNRY}_{\text{proposed}} - \text{PSNRY}_{\text{JM11.0}} \ [\text{dB}]
\]

\[
\Delta \text{Bitrate} = \frac{\text{Bitrate}_{\text{proposed}} - \text{Bitrate}_{\text{JM11.0}}}{\text{Bitrate}_{\text{JM11.0}}} \times 100\% \ [\%]
\]

\[
\Delta \text{Time} = \frac{\text{Time}_{\text{proposed}} - \text{Time}_{\text{JM11.0}}}{\text{Time}_{\text{JM11.0}}} \times 100\% \ [\%]
\]

**B. Coding Performances**

Table II shows our simulation results for the fast intra mode decision algorithms. In the table, we suppose that the positive sign indicates an increase, and the negative sign indicates a decrease. All experimental results of the proposed algorithms are compared to those of the original fast intra prediction mode decision algorithm which is implemented on the reference software JM11.0.

From Table II, our experimental results show that the proposed algorithm 1 achieves 80.82% time savings on average while ensuring a negligible loss in PSNR values and a slight increase in bitrates. The results of time savings are significant while the losses in PSNR and bitrate values are slightly decreasing 0.13 dB and acceptable increasing 4.84% on average, respectively.

Since we use the unconditional DC mode decision algorithm, the bitrate increment of the proposed algorithm 1 is a little high but acceptable. In order to reduce the bitrate increment, we refine the DC mode condition by using the condition of block boundary to select the DC mode. Hence, we propose algorithm 2 to reduce the bitrate increment by 3.42% on average while saving the encoding time and simultaneously slightly increasing the PSNR values. Hence, we can say that the propose algorithm 2 is the best algorithm in terms of bit savings compared with the proposed algorithm 1. We also obtain both a reduction of the encoding time and a slight increment in PSNR values for all test sequences by using the proposed algorithm 2.

In order to illustrate a negligible loss in PSNR values, an increment in bitrates, and a significant time saving, we show the rate-distortion (RD) and the running time curves of "Mobile" (QCIF) and "Football" (CIF) sequences. We observe similar improvements in other test sequences.

Figure 5(a) presents the RD curves of "Mobile" (QCIF) sequence. The figure indicates that those curves almost overlap, and this is especially true for the RD curve of the proposed algorithm 2 and that of JM11.0. Figure 6(a) shows the RD curves of "Football" (CIF) sequence. The RD curves of the proposed algorithms 1 and 2 overlap that of JM11.0. This means that the RD performance of the proposed algorithms is almost similar to that of the reference software JM11.0. Therefore, the proposed algorithms have a negligible rate-distortion degradation.

Figure 5(b) and Figure 6(b) show the time savings of "Mobile" (QCIF) sequence and that of "Football" (CIF) sequence, respectively. The running time curves of the two proposed algorithms lie bellow that of the original reference software JM11.0 in both test sequences. Figure 5(b) and Figure 6(b) indicate that the proposed algorithms outperform the original H.264 implemented on the reference software JM11.0 in terms of time savings.

323
C. Comparison with Other Algorithms

We compare the results of the two proposed algorithms with those of other algorithms experimented on the same sequences in Table III, i.e., the algorithms in the references [4]-[8]. The results in the table are the average results of four QP values, as shown in Table I.

From Table III, the time saving of the proposed algorithm 1 lies between 80.24% and 85.06% while those of the other algorithms lie between 55.026% and 76.87%. This means that the proposed algorithm 1 can reduces much more time than the other algorithms can. In addition, the bit rate increment resulting from the proposed algorithm 1 lies around +5% while those of resulting from the other algorithms lie up to +5.405%. From this comparison, we can say that the proposed

algorithm 1 results in similar bitrate saving to the other algorithms while outperforming the other algorithms in terms of the encoding time.

As shown in Table III, the results of the proposed algorithm 1 and the other algorithms still ensure a negligible loss in PSNR values. Besides, the PSNR values of the proposed algorithm 1 are similar to those of the algorithms in [5] and [6], and the bit rates of the proposed algorithm 1 are slightly higher than those of the other algorithms.

As shown in Tables II and Table III, compared to the existing algorithms, the proposed algorithm 1 is the best algorithm in terms of the encoding time. Therefore, we can say that the proposed algorithm 1 outperforms the other algorithms for fast intra mode decision.
<table>
<thead>
<tr>
<th>Test sequence</th>
<th>Algorithm</th>
<th>ΔPSNRY [dB]</th>
<th>ΔBitrate [%]</th>
<th>ΔTime [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard (QCIF)</td>
<td>Ref. [4]</td>
<td>-0.151</td>
<td>+2.932</td>
<td>-74.91</td>
</tr>
<tr>
<td></td>
<td>Ref. [5]</td>
<td>-0.106</td>
<td>+2.361</td>
<td>-55.026</td>
</tr>
<tr>
<td></td>
<td>Ref. [6]</td>
<td>-0.088</td>
<td>+5.405</td>
<td>-72.29</td>
</tr>
<tr>
<td></td>
<td>Ref. [7]</td>
<td>-0.068</td>
<td>+0.860</td>
<td>-63.90</td>
</tr>
<tr>
<td></td>
<td>Ref. [8]</td>
<td>-0.047</td>
<td>+0.631</td>
<td>-59.62</td>
</tr>
<tr>
<td></td>
<td>Proposed 1</td>
<td>-0.036</td>
<td>+2.830</td>
<td>-61.49</td>
</tr>
<tr>
<td></td>
<td>Proposed 2</td>
<td>+0.03</td>
<td>+4.78</td>
<td>-70.73</td>
</tr>
<tr>
<td>Silent (QCIF)</td>
<td>Ref. [4]</td>
<td>-0.186</td>
<td>+3.753</td>
<td>-74.72</td>
</tr>
<tr>
<td></td>
<td>Ref. [5]</td>
<td>-0.183</td>
<td>+3.540</td>
<td>-65.170</td>
</tr>
<tr>
<td></td>
<td>Ref. [6]</td>
<td>-0.122</td>
<td>+5.441</td>
<td>-71.21</td>
</tr>
<tr>
<td></td>
<td>Ref. [7]</td>
<td>-0.055</td>
<td>+2.661</td>
<td>-61.77</td>
</tr>
<tr>
<td></td>
<td>Ref. [8]</td>
<td>-0.039</td>
<td>+1.913</td>
<td>-57.67</td>
</tr>
<tr>
<td></td>
<td>Proposed 1</td>
<td>-0.013</td>
<td>+2.998</td>
<td>-59.45</td>
</tr>
<tr>
<td></td>
<td>Proposed 2</td>
<td>+0.05</td>
<td>+3.81</td>
<td>-45.03</td>
</tr>
<tr>
<td>Mobile (CIF)</td>
<td>Ref. [4]</td>
<td>-0.137</td>
<td>+2.260</td>
<td>-76.87</td>
</tr>
<tr>
<td></td>
<td>Ref. [6]</td>
<td>-0.211</td>
<td>+3.321</td>
<td>-72.83</td>
</tr>
<tr>
<td></td>
<td>Ref. [7]</td>
<td>-0.122</td>
<td>+1.163</td>
<td>-64.46</td>
</tr>
<tr>
<td></td>
<td>Ref. [8]</td>
<td>-0.085</td>
<td>+0.852</td>
<td>-60.59</td>
</tr>
<tr>
<td></td>
<td>Proposed 1</td>
<td>-0.036</td>
<td>+1.702</td>
<td>-59.58</td>
</tr>
<tr>
<td></td>
<td>Proposed 2</td>
<td>+0.21</td>
<td>+2.01</td>
<td>-49.69</td>
</tr>
</tbody>
</table>

In order to reduce the increment of bitrate by using the unconditional DC mode decision in the proposed algorithm 1, we refine the DC mode condition by using the condition of block boundary to select the DC mode as shown in the proposed algorithm 2. Hence, the bitrate increment of the proposed algorithm 2 results in a performance better than that of the proposed algorithm 1. Moreover, the proposed algorithm 2 always shows the best improvement in PSNR values for all compared sequences, i.e., "Coastguard" (QCIF), "Silent" (QCIF), and "Mobile" (CIF) sequences. Especially, the improvement in PSNR values is up to 0.21 dB for "Mobile" (CIF) sequence.

To this end, we verify the performance of our proposed algorithms by experimental results. First of all, we propose an intra-mode decision algorithm where the DC mode is chosen as a candidate for the best prediction mode since the DC mode appears to be the superior prediction mode among the various candidates. In order to reduce the increment of bitrate and refine the DC mode condition, we propose another fast intra prediction mode decision algorithm using a condition of block boundary to select the DC mode. In fact, the proposed algorithm 1 outperforms the other algorithms in terms of coding time while the proposed algorithm 2 reduces the bitrate increment of the proposed algorithm 1.

IV. CONCLUSIONS

In this paper, we represented qualitative DC mode conditions for fast intra prediction mode decision in H.264/AVC. Through this idea, we proposed two fast intra-mode decision algorithms using the unconditional DC mode and refined the DC mode condition by using a condition of block boundary to select the DC mode. Using the proposed algorithms, on average, we either achieved 80.82% time savings with no significant rate-distortion degradation or reduced the bitrate increment of the above-mentioned proposed algorithm while saving the encoding time and simultaneously slightly increasing the PSNR values.

REFERENCES