Adaptive CU splitting and pruning method for HEVC intra coding

Gaewon Kim and Changhoon Yim

Coding unit (CU) splitting and pruning for complexity reduction in high-efficiency video coding (HEVC) intra coding is dealt with. Adaptive determination of the threshold values for splitting and pruning in each CU is proposed based on the depth level information of neighbour CUs, whose values are determined from a fixed parameter in the previous method. Simple preconditions for splitting and pruning are also proposed to improve coding efficiency. Simulation results show that the proposed method gives significant improvement of computational complexity with much smaller reduction of coding efficiency compared with the previous method.

Introduction: The high-efficiency video coding (HEVC) standard employs a quadtree-based coding unit (CU) instead of a fixed size macroblock [1]. By adopting a quadtree structure, the HEVC provides notable improvements of coding efficiency, but increases computational complexity excessively compared with previous video coding standards [2]. A largest CU can be recursively partitioned until the CU reaches the smallest CU [1, 2]. To find the optimum combination of CUs, the HEVC test model (HM) [4] conducts rate distortion optimisation (RDO). However, the RDO operation is a major factor for excessive increase in computational complexity due to the full search for possible combinations of CUs.

Previous CU splitting and pruning method: In [3], a fast CU splitting and pruning method for intra coding was presented to reduce the computational complexity. It is conducted in two complementary steps: early CU splitting test and early CU pruning test. The early CU splitting signifies that the current CU is partitioned into four equal-sized CUs without the computation of full rate distortion (FRD) cost. The early CU pruning signifies that the current CU is not to be partitioned any longer. These steps are operated based on a probabilistic decision rule.

Before the early test steps are applied, statistical parameters have to be periodically updated. The average $\mu_{L,1}$ and standard deviation $\sigma_{L,1}$ parameters are computed from low-complexity rate distortion (LRD) costs $J_L$ for the CUs that are not partitioned (non-split). The average $\mu_{F,2}$ and standard deviation $\sigma_{F,2}$ parameters are computed from the FRD cost $J_F$ for the CUs to be partitioned (split). The probability distributions of $J_L$ and $J_F$ are assumed to be normal (Gaussian) [3].

In the encoding process for intra coding, the early CU splitting test is performed after the rough mode decision (MD) step in which the LRD costs $J_L$ are computed to select candidate modes among 35 intra prediction modes. The threshold value $T_L$ for the early CU splitting test is computed as [3]

$$T_L = \mu_{L,1} + z \sigma_{L,1}$$  \hspace{1cm} (1)

In the early splitting test, the minimum $J_L$ value is compared with $T_L$. If the minimum $J_L$ of the current CU is larger than $T_L$, then the CU is early split. If the CU is decided to be split by the early CU splitting test, the MD step is skipped. Otherwise, the early CU pruning test is performed after the MD step in which the FRD costs $J_F$ are computed to select the best prediction mode among the candidate modes. The threshold value $T_F$ for the early CU pruning test is computed as

$$T_F = \mu_{F,2} - z \sigma_{F,2} \hspace{1cm} (2)$$

In the early pruning test, the $J_F$ of the best prediction mode is compared with $T_F$. If the $J_F$ is smaller than $T_F$, the CU is early pruned. In (1) and (2), $z$ is the upper bound of the integral for the calculation of the probability of wrong decision ($\alpha$). The $\alpha$ value is given as a parameter between 0.1 and 0.5 and the $z$ value can be obtained indirectly from $\alpha$ using a cumulative standard normal distribution table [3].

As the $\alpha$ value becomes larger, the probabilities for the early CU splitting and pruning are increased. Consequently, the computational complexity would be decreased, but the bit rate would be increased. On the other hand, as the $\alpha$ value becomes smaller, the probabilities for the early CU splitting and pruning are decreased. In this case, the reduction of computational complexity would become smaller, and the increase of bit rate would become smaller.

As the image becomes more complex, the method in [3] gives relatively larger amounts of increase in bit rate in the simulation results. It is observed that the probability for a CU to be misclassified (early split or pruned incorrectly) becomes larger as the image becomes more complex. Another problem for the method in [3] is that the $\alpha$ value should be fixed for the whole sequence without the knowledge of video characteristics.

Proposed method: Fig. 1 shows the flowchart of the proposed CU splitting and pruning method.

Let $d$ represent the current depth level. Let $d_1$ and $d_2$ represent the depth levels of neighbour CUs which are the left CU and the above CU, respectively. The average depth level $\overline{d}$ of neighbour CUs is defined as

$$\overline{d} = \sum_c c_i d_i$$  \hspace{1cm} (3)

where $c_i$ is the weighting factor of the $i$th neighbour CU and the sum of $c_i$ values is 1.

We propose to determine adaptively the threshold $T_L$ and $T_F$ values based on the average depth level $\overline{d}$ of neighbour CUs. The threshold value $T_L$ for the proposed early CU splitting test is determined as

$$T_L = \mu_{L,1} + \delta (d_{\text{max}} - \overline{d}) \sigma_{L,1}$$  \hspace{1cm} (4)

where $d_{\text{max}}$ is the maximum depth level and $\delta$ is a constant which is related to the probability range of wrong decision.

As $\overline{d}$ becomes larger, it means that the neighbour CUs have been split more frequently. Since the current CU would have similar characteristics as the neighbour CUs, the probability for the current CU to be split would be smaller. In this case, $(d_{\text{max}} - \overline{d})$ becomes smaller and the current CU is controlled to be split more aggressively. If $\overline{d}$ approaches $d_{\text{max}}$, $T_L$ becomes minimised as $\delta$ becomes smaller and the current CU is controlled to split more conservatively. If $\overline{d}$ approaches a fixed value for $\alpha$ as in the previous method [3], $T_L$ becomes the maximum value, which corresponds to the case that the probability of wrong decision ($\alpha$) becomes minimised as 0.5.

The threshold value $T_F$ for the proposed early CU pruning test is determined as

$$T_F = \mu_{F,2} - \delta \overline{d} \sigma_{F,2}$$  \hspace{1cm} (5)

As $\overline{d}$ becomes smaller, the current CU is controlled to be pruned more frequently for complexity reduction. If $\overline{d}$ approaches $0$, $T_F$ becomes maximised as $\mu_{F,2}$ which corresponds to the case that $\alpha$ becomes maximised as 0.5. On the other hand, as $\overline{d}$ becomes smaller, the probability for the current CU to be split would be decreased. In this case, $(d_{\text{max}} - \overline{d})$ becomes larger and the current CU is controlled to be split more conservatively. If $\overline{d}$ approaches 0, $T_F$ approaches the maximum value, which corresponds to the case that $\alpha$ becomes minimised as 0.1.

We also propose a simple precondition for the early CU splitting as

$$\overline{d} > d + 0.5$$  \hspace{1cm} (6)

The tendency whether the CU is to be split at the current depth level $d$ is typically similar to whether neighbour CUs were split at the depth level.

Fig. 1 Flowchart of proposed CU splitting and pruning method
If $\alpha$ is relatively larger than $d$, then the CU would be split with high probability. A simple precondition for the early pruning is also proposed as

$$\tilde{d} \leq d + 0.5$$  \hspace{1cm} (7)

If $\tilde{d}$ is relatively smaller than $d$, then the CU is not likely to be split and is possible to be pruned. By including these simple preconditions as in (6) and (7) for splitting and pruning, respectively, the probabilities for incorrect early decision (misclassification) are decreased and the coding efficiency is improved.

**Simulation results:** The proposed method is implemented on HM12.1 [4] and is simulated in intra mode with QP values of 22, 27, 32 and 37 for five video sequences. The $\delta$ value in (4) and (5) is set to 0.427, then the probability range of wrong decision ($\alpha$) corresponds to [0.1, 0.5].

Table 1 shows the simulation results for five video sequences. It compares BD-rate (%), BD-peak signal-to-noise ratio (PSNR) (dB) and average time saving (ATS) for the proposed method and the previous method [3] with $\alpha = 0.1, 0.3, 0.5$. The BD-rate and BD-PSNR values are obtained relative to HM12.1 [4] using the Excel in [5]. The BD-rate (%) in Table 1 represents the increase of bit rate relative to the BD-rate is increased, it means that the coding efficiency is improved as the trade-off with the BD-rate increase. If measurement in [3] and represents the reduction rate of computation time with 35.18% of ATS, whereas the previous method gives unacceptable BD-rate with $\alpha = 0.3$ and 0.5 and unsatisfactory ATS (14.91%) with $\alpha = 0.1$. The amount of reduction of BD-PSNR in the proposed method is overall less than those in the previous method.

Hence, we conclude that the proposed method achieves substantial improvement in complexity reduction with much smaller reduction of coding efficiency in terms of BD-rate and BD-PSNR compared with the previous method.

**Acknowledgments:** This research was supported by the IT R&D programme of MSIP/KEIT (10044667, Live encoder for 4K UHD and N screen real-time broadcasting based on HEVC codec that reduce transmission bandwidth doubled compared to H.264 codec), by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0009426) and by the BK21 Plus. This work was also supported by the Korea Ministry of Land, Infrastructure and Transport as U-City Master and Doctor Course Grant Program.

---

Table 1: Simulation results for five video sequences (PS: People on Street, Ki: Kimono1, Te: Tennis, BD: Basketball Drive and Ca: Cactus)

<table>
<thead>
<tr>
<th></th>
<th>PS</th>
<th>Ki</th>
<th>Te</th>
<th>BD</th>
<th>Ca</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3] ($\alpha = 0.1$)</td>
<td>BD-rate (%)</td>
<td>2.66</td>
<td>3.27</td>
<td>5.86</td>
<td>0.65</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>BD-PSNR (dB)</td>
<td>-0.15</td>
<td>-0.11</td>
<td>-0.17</td>
<td>-0.02</td>
<td>-0.03</td>
</tr>
<tr>
<td></td>
<td>ATS (%)</td>
<td>28.03</td>
<td>14.91</td>
<td>15.52</td>
<td>13.29</td>
<td>13.57</td>
</tr>
<tr>
<td>[3] ($\alpha = 0.3$)</td>
<td>BD-rate (%)</td>
<td>7.29</td>
<td>7.65</td>
<td>12.47</td>
<td>3.58</td>
<td>4.10</td>
</tr>
<tr>
<td></td>
<td>BD-PSNR (dB)</td>
<td>-0.41</td>
<td>-0.26</td>
<td>-0.36</td>
<td>-0.10</td>
<td>-0.15</td>
</tr>
<tr>
<td></td>
<td>ATS (%)</td>
<td>51.36</td>
<td>37.95</td>
<td>53.37</td>
<td>51.32</td>
<td>44.18</td>
</tr>
<tr>
<td>[3] ($\alpha = 0.5$)</td>
<td>BD-rate (%)</td>
<td>7.23</td>
<td>11.97</td>
<td>17.30</td>
<td>5.64</td>
<td>4.32</td>
</tr>
<tr>
<td></td>
<td>BD-PSNR (dB)</td>
<td>-0.41</td>
<td>-0.41</td>
<td>-0.50</td>
<td>-0.16</td>
<td>-0.16</td>
</tr>
<tr>
<td></td>
<td>ATS (%)</td>
<td>56.33</td>
<td>54.28</td>
<td>59.44</td>
<td>55.16</td>
<td>48.71</td>
</tr>
<tr>
<td>Proposed</td>
<td>BD-rate (%)</td>
<td>2.63</td>
<td>1.86</td>
<td>2.71</td>
<td>2.39</td>
<td>2.26</td>
</tr>
<tr>
<td></td>
<td>BD-PSNR (dB)</td>
<td>-0.15</td>
<td>-0.07</td>
<td>-0.08</td>
<td>-0.07</td>
<td>-0.08</td>
</tr>
<tr>
<td></td>
<td>ATS (%)</td>
<td>31.49</td>
<td>35.18</td>
<td>43.28</td>
<td>35.74</td>
<td>28.44</td>
</tr>
</tbody>
</table>

When $\alpha = 0.5$ in the previous method [3], the average ATS value is satisfactory (54.78%), but the BD-rate is unacceptable large (9.29%). The $\alpha = 0.3$ case also gives unacceptable large average value of BD-rate (7.02%). For acceptable average BD-rate, the previous method needs to select the $\alpha$ value conservatively as 0.1, which gives the average BD-rate of 2.66%. However, the ATS is not satisfactory (17.06%) in this case. For some video sequence (e.g. BD), the previous method gives acceptable BD-rate with $\alpha = 0.3$. However, for other video sequences (e.g. Ki and Te), the previous method gives acceptable BD-rate only with $\alpha = 0.1$. It is hard to select the appropriate fixed value for $\alpha$ without the knowledge of video characteristics.

---

© The Institution of Engineering and Technology 2014
21 February 2014
doi: 10.1049/el.2014.0647
One or more of the Figures in this Letter are available in colour online.
Gaewon Kim and Changhoon Yim (Department of Internet and Multimedia Engineering, Konkuk University, Seoul, Republic of Korea)
E-mail: cyim@konkuk.ac.kr

References
4 HM reference software 12.1 [Online]. Available at https://www.hevc.hhi.fraunhofer.de/trac/hevc/browser/tags