|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, Switzerland, 21-30 July, 2011 | Document: JCTVC-G156\_r1  WG11 number: m21709 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CU Depth Pruning for Fast Coding Tree Block Decision** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Hui Li Tan  Chuohao Yeo  Yih Han Tan  1 Fusionopolis Way  #21-01 Connexis (South Tower)  Singapore 138632 | Tel: Email: | +65 6408 2000  chyeo@i2r.a-star.edu.sg |
| *Source:* | Institute for Infocomm Research, A\*STAR | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This contribution presents a CU Depth Pruning algorithm for fast coding tree block (CTB) decision. The proposed method attempts to terminate the CTB decision process by performing a one-level look-ahead for the last sub-CU where possible. It is reported that the proposed method reduces encoding time by about 8% with 0.1% Luma BD-Rate coding loss for the Random Access and Low Delay configurations.

# Introduction

In HEVC, coding block tree (CTB), based on a recursive quad-tree decomposition of each largest coding unit (LCU), is used for the representation of variable block sizes, so that regions of different sizes can be better coded. Regions of high stationarity and homogeneity can possibly be coded with a larger block size, which results in a smaller side-information overhead. However, the flexibility of the variable block size structure greatly increases the search domain and hence the computational complexity of RDO at the encoder. To speed up the CTB and mode decision process, one possible approach is to reduce the search space by avoiding full branching of the search tree by making early CTB termination.

# Background

For ease of discussion, we shall use the following notation. Referring to Figure 1, each frame is divided into non-overlapping LCUs, where denotes the *i*0th LCU within the frame. Let *M* denotes the maximum depth of the CTB representation, and *m*0 denotes a parameter for deciding the minimum CU size. Then, a CU at depth *m*, where 0 ≤*m*<*M*, is of size; therefore the LCU size iswhile the minimum CU size is. In the HEVC common test conditions [1], *M*=4 and *m*0=2, i.e., the LCU size is 64x64, and the minimum CU size is 8x8.

Each CU at depth *m* is denoted by, where *i*0 indexes the location of the root LCU within the frame, while each subsequent index, *i*1, …, *im* (0≤*i*1, …, *im*≤3), specifies the index of the CU within its parent. Therefore, is the *im*th CU of. This notation would allow us to uniquely identify each CU within a frame. For convenience, we would also sometimes use to denote, i.e., denotes the list of indices *i*0, *i*1, …, *im*.

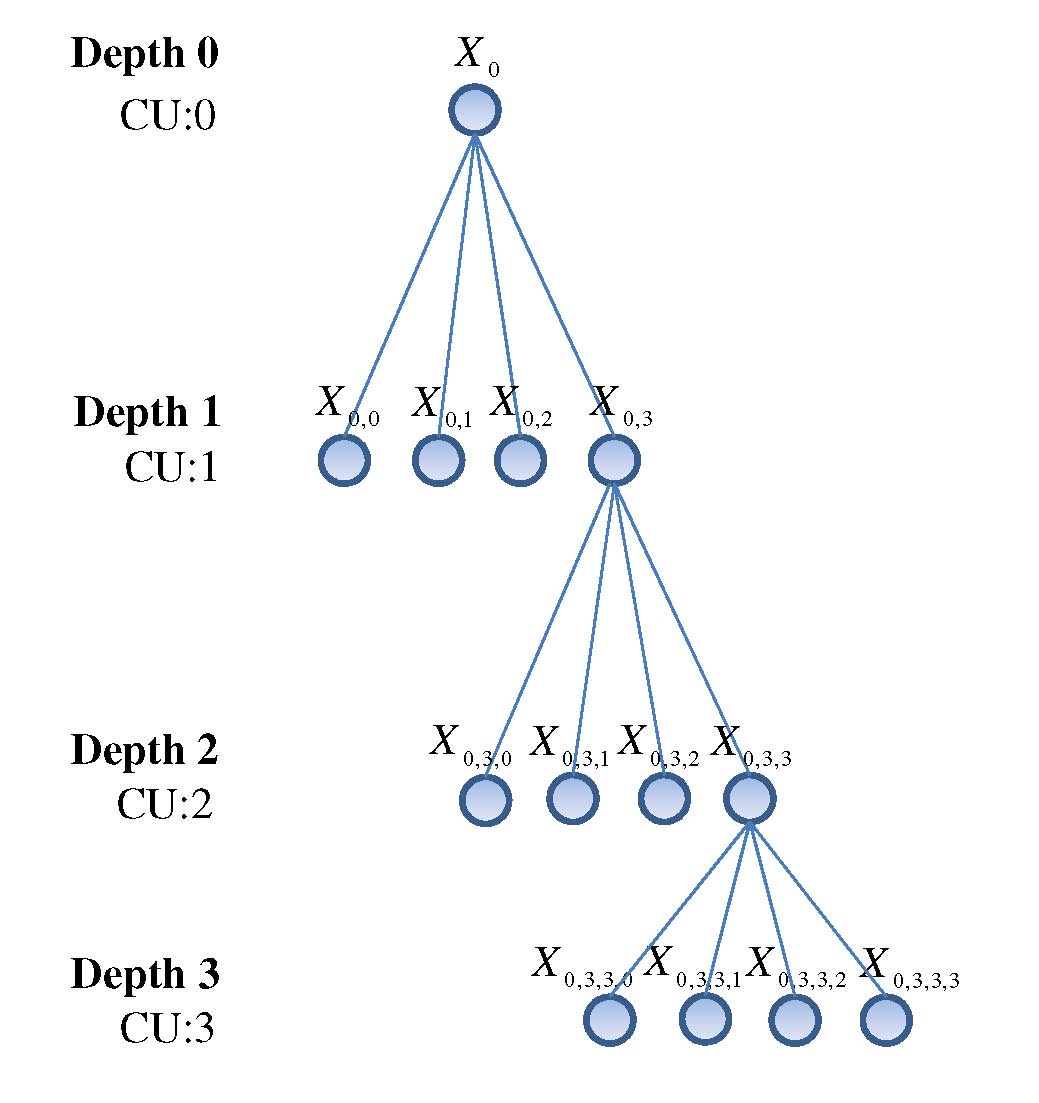


Figure 1. Coding Tree Blocks (CTB).

Denote F(to be the best RD cost computed for the CU, , assuming that is not split into sub-CUs. Denote C( to be the best RD cost computed for, without any restriction on whether it is split or not.

In the HM4 reference software encoder, the mode decision process for each CU is carried out as follows. Its best coding mode and PU partition type are first determined assuming that the CU is not split, before recursively repeating the same process for each of its sub-CUs. A final decision is then made on whether the CU is to be split or not only after all the sub-CUs has been analyzed. Therefore, the following recursive relationship is used to determine in the HM4 reference software encoder:

where *C0* and *C1* are the overhead of not splitting the CU and splitting the CU respectively.

# Proposed method

Since the CTB is traversed in a depth-first manner during CU mode decision, for each CU that is the 4th sub-CU of its parent, i.e. , the best RD cost without restriction of splitting for its previous three siblings sub-CUs, i.e.are known and can be used as inputs for the early CTB termination condition for . In the proposed method, for each CU that is the 4th sub-CU of its parent, , if the sum of the RD cost of its sibling sub-CUs and itself is larger than the best current RD cost of its parent CU, i.e., if the following holds:

,

CU splitting is terminated for .

The intuition behind this is that splitting the CUs typically improves prediction at the cost of increasing overhead, and if a partial split of the CU does not result in any improvement in RD cost, then it is unlikely that a CTB which splits the current CU would result in a better RD cost.

In our implementation, this check is done only at depth *m*=1 (32x32) and *m*=2 (16x16), since at depth *m=3* (8x8), the sub-CU cannot be split any further.

A patch for the source code modifications proposed is attached. Note that the patch also includes modifications for handling an additional encoder configuration flag.

# Experimental results

The proposed method was implemented based on HM4.0. Tests were conducted following the common testing conditions [1].

Table 1 shows the performance of the proposal.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.2% | -0.3% | -0.3% | 0.1% | 0.1% | 0.1% |
| Class B | 0.1% | -0.2% | -0.3% | 0.1% | 0.1% | 0.1% |
| Class C | 0.1% | -0.1% | -0.2% | 0.0% | 0.0% | 0.0% |
| Class D | 0.0% | -0.1% | -0.1% | 0.0% | 0.0% | 0.0% |
| Class E | 0.2% | -0.6% | -0.3% | 0.1% | 0.0% | 0.1% |
| **Overall** | 0.1% | -0.3% | -0.2% | 0.1% | 0.0% | 0.1% |
|  | 0.1% | -0.2% | -0.2% | 0.1% | 0.0% | 0.1% |
| Enc Time[%] | 96% | | | 94% | | |
| Dec Time[%] | 98% | | | 97% | | |
|  |  |  |  |  |  |  |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.1% | -0.3% | -0.1% | 0.1% | 0.6% | 0.1% |
| Class B | 0.1% | -0.1% | -0.4% | 0.1% | 0.1% | 0.1% |
| Class C | 0.1% | 0.0% | 0.0% | 0.1% | 0.1% | 0.1% |
| Class D | 0.1% | -0.1% | 0.0% | 0.0% | -0.2% | 0.0% |
| Class E |  |  |  |  |  |  |
| **Overall** | 0.1% | -0.1% | -0.1% | 0.1% | 0.2% | 0.1% |
|  | 0.1% | -0.2% | -0.1% | 0.1% | 0.1% | 0.1% |
| Enc Time[%] | 92% | | | 90% | | |
| Dec Time[%] | 97% | | | 96% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.1% | 0.0% | -0.3% | 0.1% | 0.1% | 0.3% |
| Class C | 0.1% | -0.1% | -0.1% | 0.1% | 0.3% | -0.2% |
| Class D | 0.1% | -0.1% | 0.3% | 0.1% | 0.3% | -0.1% |
| Class E | 0.1% | -1.0% | 0.0% | 0.1% | -0.1% | -0.1% |
| **Overall** | 0.1% | -0.2% | -0.1% | 0.1% | 0.1% | 0.0% |
|  | 0.1% | -0.2% | -0.1% | 0.1% | 0.1% | 0.0% |
| Enc Time[%] | 92% | | | 92% | | |
| Dec Time[%] | 97% | | | 98% | | |

Table 1. Results of applying CU Depth Pruning.

# Conclusion

CU Depth Pruning reduces encoding time by about 8% with 0.1% coding loss for the RA and LB configurations. For the All Intra cases, encoding time is reduced by about 5% with 0.1% coding loss. We recommend including this technique as an encoder configurable tool in the next release of the HEVC reference software.

# Reference

1. F. Bossen, “Common conditions and software reference configurations,” JCTVC-F900, Torino, Italy, Jul., 2011.

# Patent rights declaration(s)

**Institute for Infocomm Research may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**