|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  6th Meeting: Torino, 14-22 July, 2011 | Document: JCTVC-F543 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Tableless run-length coding for transform coefficients in CAVLC** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Antti Hallapuro, Jewon Kang, Jani Lainema, Kemal Ugur | Email: | [antti.hallapuro@nokia.com](mailto:antti.hallapuro@nokia.com)  [ext-jewon.kang@nokia.com](mailto:ext-jewon.kang@nokia.com)  [jani.lainema@nokia.com](mailto:jani.lainema@nokia.com)  [kemal.ugur@nokia.com](mailto:kemal.ugur@nokia.com) |
| *Source:* | Nokia | | |

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

# Abstract

The contribution proposes a tableless algorithm for run-length coding of transform coefficients in CAVLC. A simple equation is used for generating the run-length values with special emphasis on the longest run-length value. As a result, it is reported that up to 812 bytes of memory required by the HM3 design can be saved in both encoder and decoder. The contribution reports average BD bitrate impact of   
-0.1% for intra LC, -0.1% for random access LC and +0.1% for low delay LC configurations.

# Introduction

It has been reported that large run-length tables of the HM3 CAVLC design are expensive in hardware implementations and several contributions have tried to solve the problem. Currently, the table is used for lowest 28 scan positions and the table has the size of 28\*29=812 bytes (or 434 bytes if redundant entries are not counted).

This contribution solves the memory size problem by completely removing the run-length table from the CAVLC branch and replacing it with a simple equation.

# Algorithm

The basic principle of the algorithm is that most of the runs can be sequentially mapped to codeword numbers (i.e. codeword number is set equal to run value). On top of that, the longest run can be more carefully dealt with because of its frequent occurrences in the run-length coding. That is, the code number of the longest run can be moved forward for more efficient coding. Based on the observed probability distribution of the run-lengths, the probability of the longest run is mostly between the shortest and the second longest run. Hence, a simple linear equation is created for defining the position of longest run, which works reasonably well:

position\_of\_longest\_run = (maxrun + A) / K

where maxrun is the maximum run for a scan position minus 1 as used in HM. K is chosen so that division can be implemented with bit shift. The proposed algorithm can be implemented in the decoder as follows.

Given the values of maxrun and decoded codenumber (cn), the run value and level indication are decoded as:

if(cn<maxrun+2)

{

level = 0; // next non-zero level is 1

thr = (maxrun + 1) >> scale;

if (cn >= thr)

run = (cn == thr) ? (maxrun+1) : (cn-1);

else

run = cn;

}

else

{

level = 1; // next non-zero level is larger than 1

run = cn-maxrun-2;

}

The parameter *scale* is set to 0 for intra chroma blocks and 3 for inter luma inter and inter chroma blocks.

The proposed method is not used for intra luma coefficients since those coefficients are coded in HM3 using a separate method that does not utilize run-length table.

# Simulation results

Simulation results were obtained with HM3.0 software. The results show BD bitrate difference of -0.1% for all intra, -0.1% for random access and +0.1% for low delay LC configurations. Detailed results are shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Intra LC | | |
| Y BD-rate | U BD-rate | V BD-rate |
| Class A | -0,1 | -0,3 | -0,2 |
| Class B | -0,1 | -0,4 | -0,3 |
| Class C | -0,2 | -0,8 | -0,9 |
| Class D | -0,2 | -0,8 | -1,0 |
| Class E | -0,1 | 0,3 | 0,0 |
| All | -0,1 | -0,4 | -0,5 |
| Enc Time[%] | 99 % | | |
| Dec Time[%] | 100 % | | |
|  |  |  |  |
|  |  | Random access LC |  |
| Y BD-rate | U BD-rate | V BD-rate |
| Class A | 0,0 | 0,1 | 0,2 |
| Class B | -0,1 | 0,3 | 0,3 |
| Class C | -0,1 | -0,3 | -0,4 |
| Class D | -0,1 | -0,4 | -0,4 |
| Class E |  |  |  |
| All | -0,1 | 0,0 | -0,1 |
| Enc Time[%] | 99 % | | |
| Dec Time[%] | 99 % | | |
|  |  |  |  |
|  |  |  |  |
|  | Low delay LC | | |
|  | Y BD-rate | U BD-rate | V BD-rate |
| Class A |  |  |  |
| Class B | 0,1 | 0,4 | 0,1 |
| Class C | 0,1 | 0,2 | 0,1 |
| Class D | 0,1 | 0,3 | 0,5 |
| Class E | 0,4 | 2,1 | -0,9 |
| All | 0,1 | 0,6 | 0,0 |
| Enc Time[%] | 100 % | | |
| Dec Time[%] | 100 % | | |

# Conclusions

A tableless run-length coding method was presented. The proposed algorithm saves up to 812 bytes of memory in CAVLC. This is accomplished by replacing the run-length table with an easy to compute equation. The experimental results indicate that there is no significant loss in coding performance due to this simplification.

# Patent rights declaration(s)

**Nokia corporation may have IPR 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).**