|  |  |
| --- | --- |
| Joint Collaborative Team on Video Coding (JCT-VC)  of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11  16th Meeting: San José, US, 9–17 Jan. 2014 | Document: JCTVC-P0115 |

|  |  |  |  |
| --- | --- | --- | --- |
| Title: | Non-RCE4: Transition copy mode for Palette mode | | |
| Status: | Input Document to JCT-VC | | |
| Purpose: | Proposal | | |
| Author(s) or Contact(s): | C. Gisquet, G. Laroche, P. Onno  Canon Research Centre France Rue de la Touche Lambert 35510 CESSON-SEVIGNE, FRANCE | Tel: Email: | +33 299876800 [christophe.gisquet@crf.canon.fr](mailto:christophe.gisquet@crf.canon.fr) [patrice.onno@crf.canon.fr](mailto:patrice.onno@crf.canon.fr) |
| Source: | Canon | | |

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

# Abstract

In the scope of this meeting, RCE4 studies several proposals related to palette-like coding methods. In particular, Test2 related to contribution JCTVC-O0218 [1] evaluates a palette coding tool which predicts indices using the above line or the left index. It is proposed in the present contribution to add new prediction modes that take into account the last occurring position of indices. It is reported that the use of one such prediction mode provides gains for respectively AI/RA/LDB configurations, on top of RCE4 Test2, of more than 3.2%/3.0%/1.3% for SC classes, and up to 7.5%/7.2%/5.7% on optional classes.

# Introduction

RCE4 test2 proposes 2 index predictions methods that can be seen in the following figure:

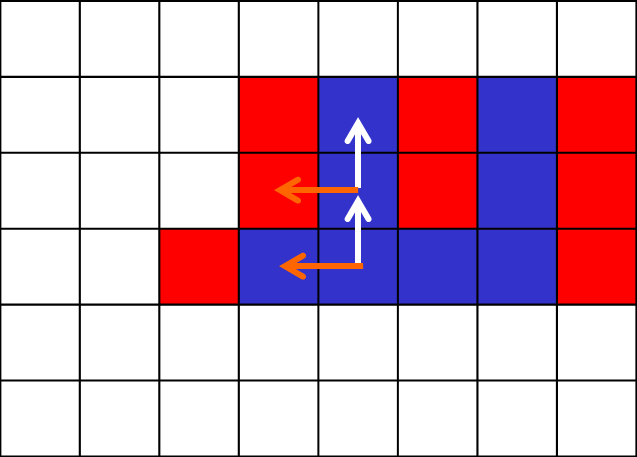


Figure 1 : index prediction modes in RCE4 test2

In the above example, we have represented arrows for these 2 prediction modes:

* The orange arrow copies from the left, i.e. represents a repetition of the same value;
* The white arrow copies from the above, i.e. is a repetition of the line above.

In the above example, one can easily deduce that the above mode will be selected for the first case, while the left mode will be used for the second one.

These modes are well suited for strictly vertical or horizontal features. However, there are instances where palette-like content exhibits different behaviour. In particular, one such content can be observed below:

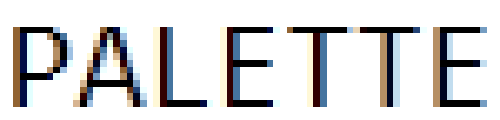


Figure 2 : example of specific palette content

This example represents text rendered with antialiasing, which is critical to user reading comfort. In such an example, one can clearly see the existence of so-called transitions. One can easily notice the repetitiveness of these transitions: the red, orange and blue arrows show repetitive transitions across the above figure.

# Proposal

## Concept

We propose adding a new prediction mode, called here transition mode, which takes into account above transitions. The idea is, for each palette index, to record the position of its last occurrence. Then, the new prediction mode will use these positions as start places for copying, depending on the value of the pixel on the left.

The above can be represented in the following figure:

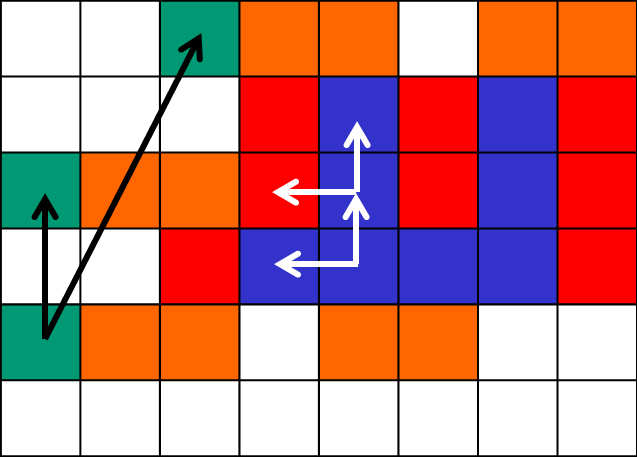


Figure 3 : transition copy mode

Classical prediction modes are identical here, while we have presented the new possible modes with black arrows. In these cases, when encoding the orange index, the green one on the left is an indication of a potential transition. Now, having recorded last positions of the green index, it is possible to copy from these positions. The above example is a simplified version, but Figure 2 easily shows how that can apply.

## Algorithm

The modified algorithm for coding the index can be represented as follows:

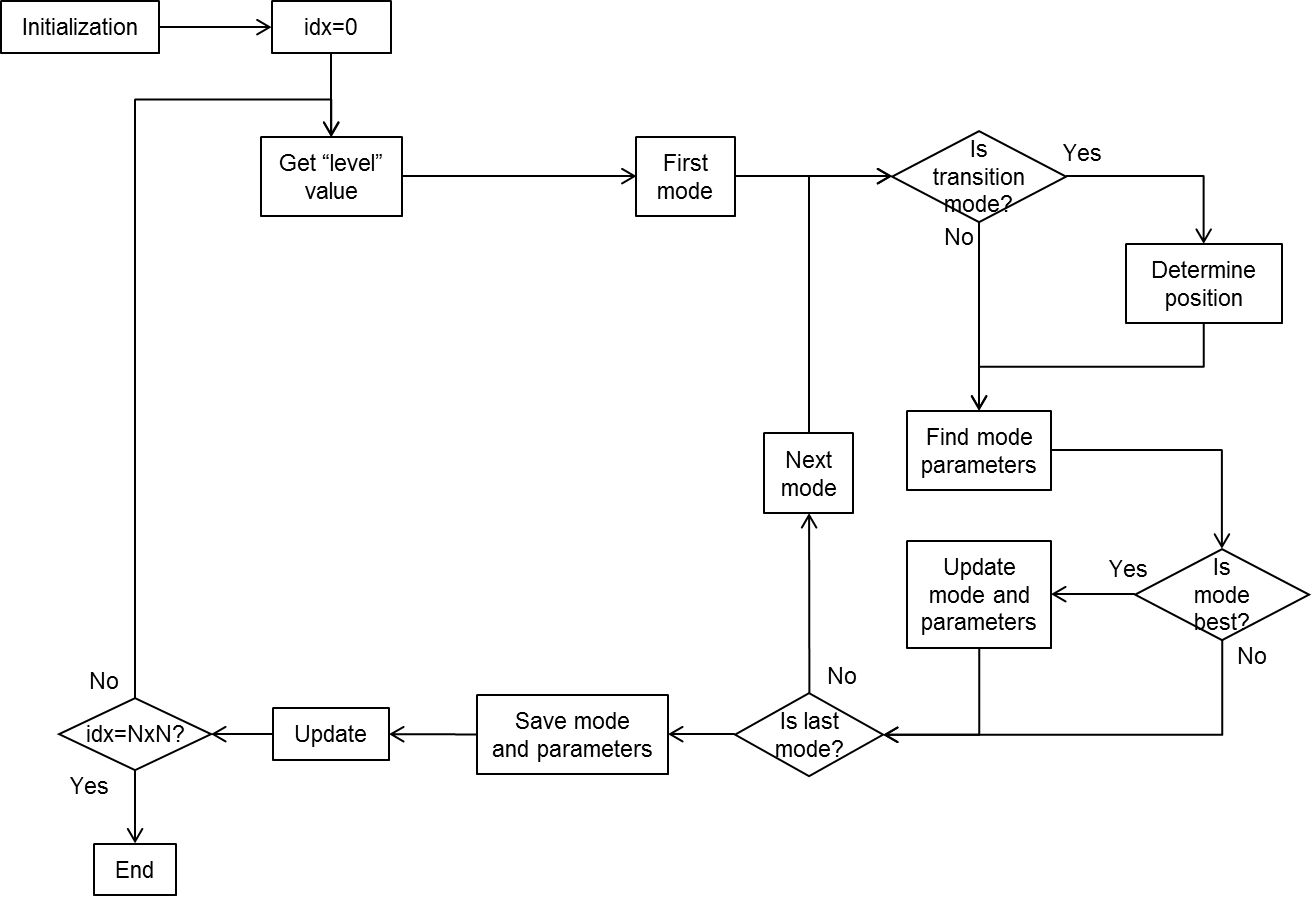


Figure 4 : encoder algorithm

The new transition mode can operate mostly similarly to the copy from above mode, except the position from which to copy is not, for a CU of size NxN, the position idx-N above but a position which depends on the last occurrence of the level at the left of the current position.

## Remarks on the position storage

The standardization of the BDC-1 algorithm in the VESA task group for display stream compression, where a classical Lempel-Ziv algorithm with a 16KB dictionary is used to compress input streams of up to 17.3Gbits/s shows that such a memory is achievable at low costs, as it is planned to be used in DisplayPort and HDMI connections.

However, the new prediction mode increases the memory requirement from 64\*5 bits to 4095\*5 bits (less than 2.5KB). We have therefore studied the impact of restricting the available memory, two results will be presented, using either the whole CU or only the P=26=64 last levels. Let us call P the maximal distance allowed, obviously a power of 2 here.

The most obvious way to achieve this is, for instance:

* Not testing the transition mode when it would copy away from a position resulting in a distance more than P;
* Using another default position, e.g. use idx-N, or force the position to be idx-P.

Instead, when the whole CU cannot be used, we have chosen to only store the LSBs of the positions. The used position can be reconstructed by using the MSBs of the current position and these LSBs. When this determined position overflows the current position (idx), simply subtracting P from the position allows remaining within P positions of the current position. It can be noted that henceforth, the position now requires, for P=64, 6 bits instead of a worst-case of 12.

## Entropy coding

Only the mode coding has been modified. It currently uses a flag whose context is based on a neighbor mode. We have extended this by using the following codewords:

|  |  |
| --- | --- |
| Codeword | Mode |
| 0 | Copy left |
| 10 | Copy transition |
| 11 | Copy above |

Table 1 : prediction mode codewords

# Results

## CU mode

In that case, we are using one transition mode.

Using REXT5.1 as reference, we have:







Using RCE4 Test2 as reference, so as to better visualize the relative improvement, we have:







## Last 64 elements mode

Compared to RCE4 Test2, the results are:



## Informative result: use of 2 transition modes

We have evaluated using 2 transition modes, for the last 2 occurrences of each level. The obtained gains, compared to RCE4 Test2, are:



Which is around a 0.7% gain for the SC classes over 1 mode.

# Conclusion

In light of the contained complexity increase and important gains, we suggest further studying the method in a palette RCE.

# References

1. L. Guo, M. Karczewicz, J. Sole, R. Joshi (Qualcomm), “Evaluation of Palette Mode Coding on HM-12.0+RExt-4.1”, JCTVC-O0218, 15th Meeting: Geneva, CH, 23 Oct – 1 Nov. 2013

# Patent rights declaration(s)

**Canon Research Centre France 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).**