3D-CE5.h: Reducing the coding cost of merge index by dynamic merge index re-allocation

Thomas Guionnet, Laurent Guillo, Christine Guillemot

Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11
2nd Meeting: Shanghai, CN, 13–19 Oct. 2012

Document: JCT3V-B0078

Title:	**CE5.h: Reducing the coding cost of merge index by dynamic merge index re-allocation**
Status:	Input Document
Purpose:	Proposal

Author(s) or Contact(s): Thomas Guionnet, Laurent Guillo, Christine Guillemot
Email: Thomas.Guionnet@inria.fr, Laurent.Guillo @inria.fr, Christine.Guillemot@inria.fr

Source: INRIA, France

Abstract

HEVC implements a candidate vector list for merge and skip modes. When merge or skip modes are selected, a merge index is written in the bitstream. This index is first binarized using a unary code, then CABAC encoded. A CABAC context is dedicated to the first bin of the unary coded index while the remaining bins are considered as equiprobable. This strategy is efficient as long as the candidate list is constructed such as being ordered by decreasing index occurrence probability. In the context of 3D video encoding, an inter-view motion vector predictor is added at the first position of the candidate list. It is reported in this document that the inter-view motion vector predictor is not always the most probable candidate. It actually depends on the video sequence characteristics. Therefore, a dynamic candidate vector list ordering is proposed. Coding gains of 0.4 % on average are observed on side views and up to 1.0% is attained for the Ghost Town Fly sequence for both side views.

1 Introduction

HEVC relies on a candidate vector list for merge and skip modes. The efficiency of this approach depends on two parts: first the relevance of the vectors presents in the list and second the encoding efficiency of the selected merge index. The candidate vector list is designed such that the vectors are ordered by decreasing likelihood of selection. This property is then exploited for merge index coding. The merge index is binarized using a unary code, then CABAC encoded. Except for the first bin, the bins are considered as equiprobable when fed to the CABAC. Therefore, the main factor ensuring efficient merge index coding is the unary code associated to the decreasing likelihood ordering of the vectors.

In the context of 3D video coding, an inter-view motion vector predictor is added into the candidate list. This vector is especially relevant and thus selected very often. Therefore, it is always placed at first position in the candidate list. By doing simple statistics, it has been observed that although this vector is the most likely to be selected on average, its relevance depends on pictures or even sequences characteristics. Therefore, a dynamic list reordering process is proposed in order to improve the efficiency of inter-view motion vector prediction.

This document is the follow up of JCT3V-A0133 [2], as part of the core experiment CE5.h. A cross-check is available in JCT3V-B0097.

2 Proposed method

2.1 *Dynamic merge index principle*

The proposed process is performed symmetrically in the encoder and decoder. It is described on the decoder side.

For simplification purpose, the proposed method is applied only for merge index values below 3. A Merge index histogram is computed on the fly. In order to avoid constraining dependencies and robustness loss, the process is reinitialized at the same points as the entropy coder (for instance at the beginning of each slice). The histogram is not initialized with 0 values, but with arbitrary exponentially decreasing values as shown in Table 1. It brings stability when the process begins by avoiding taking index swapping decision without statistically significant data accumulation. Given a current histogram, a conversion table can be simply calculated. It allows deriving the merge index to encode given the actual index in the list, and conversely the actual index in the list given a decoded index.

Table 1: Initialization values of the merge index histogram

32	16	8

Table 2: Initialization values of the conversion table

0	1	2

Each time a Merge index is decoded from the bitstream, it is converted to its actual position in the list and the histogram is updated accordingly. The conversion table is updated only at the end of each LCU.

2.2 *Dynamic merge index process*

The method can be summarized as follows:

Slice decoding process: associate a merge index histogram array (merge_idx_histo) and a conversion table (merge_idx_conv) to each starting slice decoding process. The merge_idx_histo and merge_idx_conv arrays are initialized as depicted in Table 1 and Table 2 respectively.

Merge index parsing process: for each parsed merge_idx, set

- merge_idx = merge_idx_conv [merge_idx]
- merge_idx_histo[merge_idx] = merge_idx_histo[merge_idx] + 1

Coding unit decoding process: At the end of each LCU decoding process, update merge_idx_conv according to the current state of merge_idx_histo. An example of merge index histogram and its associated conversion table are provided in Table 3 and Table 4 respectively.

Table 3: Initialization values of the merge index histogram

40	100	60

Table 4: Initialization values of the conversion table

1	2	0

2.3 Parsing process adaptation to dynamic merge index

The merge index is binarized using the truncated unary (TU) binarization process. Up to 5 bins can be generated. The first bin is encoded using a CABAC context while the remaining bins are considered as equi-probable (bypass).

When using dynamic merge index, index re-allocation can happen at any moment. It obviously impacts the statistics of the first bin, which is encoded using CABAC. Therefore, we define a set of CABAC contexts for the first bin, one for each possible permutation of indexes. Thus, we have 6 different CABAC contexts for the first bin, adressed by the state of merge_idx_conv.

Moreover, as the dynamic merge index strategy tends to affect significantly the 3 first bins statistics, CABAC contexts are proposed for up to 3 bins. Therefore, for each of these 3 bins, 6 CABAC contexts are defined in order to take into account all possible merge index permutations. The initialisation data for all these contexts is provided in Table 6. The usage of these contexts is summarized in Table 5 and Table 7.

Table 5: Association of ctxIdx and syntax elements for each slice type in the initialisation process of merge_idx.

merge_idx_conv combination	Init type		
	0	1	2
[0, 1, 2]		0..2	18..20
[0, 2, 1]		3..5	21..23
[1, 0, 2]		6..8	24..26
[1, 2, 0]		9..11	27..29
[2, 0, 1]		12..14	30..32
[2, 1, 0]		15..17	33..35

Table 6: Values of variable initValue for merge_idx ctxIdx.

Initialisation variable	merge_idx ctxIdx																	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
initValue	122	138	153	122	138	153	122	138	153	122	138	153	122	138	153	122	138	153
Initialisation variable	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**
initValue	137	139	154	137	139	154	137	139	154	137	139	154	137	139	154	137	139	154

Table 7: Assignment of ctxIdxInc to merge_idx syntax element with context coded bins

binIdx					
0	**1**	**2**	**3**	**4**	**>=5**
0,3,6,9,12,15,18, 21,24,27,30,33	1,4,7,10,13,16,19 ,22,25,28,31,34	2,5,8,11,14,17,20 ,23,26,29,32,35	bypass	bypass	na

3 Experimental results

The experiment has been conducted with HTM 4.0.1 and evaluation is based on common test conditions [1].

Two tests were run:

- the first one is related to the integration in HTM 4.0.1 of [2] described in section 2.1 and 2.2.
- the second one implement in addition the strategy described in section 2.3 which takes advantage of CABAC contexts for the first three bins to take into account all possible merge index permutations

Results are respectively provided in Table 8 and Table 9.

Table 8 : Results obtained with HTM 4.0.1 and dynamic merge index, compared to HTM 4.0.1.

	video 0	video 1	video 2	video only	synthesized only	coded & synthesized	enc time	dec time
Balloons	0,0%	-0,1%	-0,1%	0,0%	-0,1%	-0,1%	100,7%	102,9%
Kendo	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	101,1%	103,0%
Newspapercc	0,0%	-0,1%	0,0%	0,0%	0,1%	0,1%	100,1%	101,7%
GhostTownFly	0,0%	-0,2%	-0,3%	-0,1%	0,0%	-0,1%	102,1%	102,5%
PoznanHall2	0,0%	0,1%	-0,1%	0,0%	0,0%	0,0%	102,4%	104,1%
PoznanStreet	0,0%	-0,7%	-0,3%	-0,1%	-0,1%	-0,1%	100,8%	102,0%
UndoDancer	0,0%	-0,2%	-0,3%	-0,1%	-0,1%	-0,1%	101,7%	101,1%
1024x768	0,0%	-0,1%	0,0%	0,0%	0,0%	0,0%	100,6%	102,5%
1920x1088	0,0%	-0,3%	-0,2%	-0,1%	-0,1%	-0,1%	101,7%	102,4%
Average	**0,0%**	**-0,2%**	**-0,1%**	**0,0%**	**0,0%**	**0,0%**	**101,3%**	**102,5%**

Table 9 : Results obtained with HTM 4.0.1 and dynamic merge index followed by parsing adaptation and CABAC context for the first three bins

	video 0	video 1	video 2	video only	synthesized only	coded & synthesized	enc time	dec time
Balloons	0,0%	-0,2%	-0,2%	-0,1%	-0,1%	-0,1%	100,4%	100,6%
Kendo	0,0%	-0,1%	0,0%	0,0%	0,0%	0,0%	100,6%	100,9%
Newspapercc	0,0%	-0,1%	-0,3%	-0,1%	-0,1%	-0,1%	100,0%	98,9%
GhostTownFly	0,0%	-1,0%	-1,0%	-0,2%	-0,2%	-0,2%	102,3%	99,7%
PoznanHall2	0,0%	-0,1%	-0,4%	-0,1%	-0,1%	-0,1%	102,3%	98,3%
PoznanStreet	0,0%	-0,6%	-0,4%	-0,2%	-0,2%	-0,2%	99,8%	99,6%
UndoDancer	0,0%	-0,7%	-0,6%	-0,2%	-0,2%	-0,2%	100,1%	99,7%
1024x768	0,0%	-0,1%	-0,1%	-0,1%	0,0%	0,0%	100,3%	100,1%
1920x1088	0,0%	-0,6%	-0,6%	-0,2%	-0,2%	-0,2%	101,1%	99,3%
Average	**0,0%**	**-0,4%**	**-0,4%**	**-0,1%**	**-0,1%**	**-0,1%**	**100,8%**	**99,7%**

4 Conclusion

The coding cost of merge index can be significantly reduced by adopting a dynamic merge index reallocation strategy which takes into account all possible merge index permutations, followed by an adapted parsing and context-based CABAC coding strategy. No complexity is added. We propose to adopt it.

5 References

[1] Dmytro Rusanovskyy, Karsten Müller, Anthony Vetro, « Common test conditions for 3DV Core Experiments », JCT3V-A1100, July 2012, Stockholm, Sweden.

[2] Thomas Guionnet, Laurent Guillo, Christine Guillemot, « CE5.h related: Reducing the coding cost of merge index by dynamic merge candidate list re-ordering », JCT3V-A0133, July 2012, Stockholm, Sweden.

6 Patent rights declaration(s)

INRIA 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).