CE5.h related: Merge candidate list extension for disparity compensated prediction

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
1st Meeting: Stockholm, SE, 16–20 July 2012

Document: JCT2-A0134

Title: **CE5.h related: Merge candidate list extension for disparity compensated prediction**

Status: Input Document

Purpose: Proposal

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

Source: INRIA, France

Abstract

HEVC implements a candidate vector list for merge and skip modes. The construction of this list has been extensively studied in the JCT-VC group (see for instance JCTVC-G039). It has been shown in JCTVC-I0293 that it is possible to improve the HEVC coding performance by adding in the merge list copies of the first candidate shifted by an arbitrary offset. The same basis is considered in this document and applied to disparity compensation. A gain of 0.4 % is obtained on average on side views.

1 Introduction

HEVC relies on a candidate vector list for merge and skip modes. The efficiency of this approach depends highly on the relevance of the vectors present in the list. The construction of this list has been extensively studied [1]. During the last JCT-VC meeting, it has been shown in [2] that it is possible to improve the HEVC coding performance by adding in the merge list copies of the first candidate shifted by an arbitrary offset. More precisely, in [2], if there is available space in the candidate vector list, up to two new candidates are added: the first vector of the list shifted by an offset of (-4; 0) and (4; 0). It corresponds to a horizontal shift of -1 and 1 pixel respectively. In the context of video encoding of multiple rectified views, having a fine horizontal adjustment may be desirable for efficient disparity compensated prediction. Therefore, it is proposed to extend the candidate vector list with candidates pointing to the base view and shifted by horizontal offsets.

2 Proposed method

Let us remind that a candidate in the merge list may contain one or two vectors (L0, L1 or both). The candidate list construction proceeds as follows.

1. Construct the list as currently specified.
2. Scan the list from the first to the last position. The first candidate found in the list which has either its L0 or L1 vector (or both) pointing to the base view is selected. If no such candidate exists, the process ends.
3. Derive two refined candidates by adding two offsets to the selected candidate as follows
    a. If the L0 vector is pointing to base view, add the offsets to the L0 vector only
    b. Else if the L1 vector is pointing to base view, add the offsets to the L1 vector
4. Add the two refined candidates at the end of the list. Extend the size of the list if necessary.

The offsets considered are (-4; 0) and (4; 0), as in [2]. Note that the method is valid for both uni-directional and bi-directional vectors. In the latter case, the offsets are applied to only one of the two vectors.

3 Experimental results

The experiment has been conducted with HTM 3.0 and evaluation is based on common test conditions [3].

Results are provided in Table 1.

Table 1: Results obtained with HTM 3.0 and Merge list extension with DCP candidate, compared to HTM 3.0.

	video 0	video 1	video 2	video only	synthesized only	coded & synthesized	enc time	dec time
Balloons	0,0%	-0,2%	-0,5%	-0,2%	0,0%	0,0%	98,4%	100,2%
Kendo	0,0%	-0,3%	-0,7%	-0,2%	-0,2%	-0,2%	100,1%	102,8%
Newspapercc	0,0%	-0,9%	-0,3%	-0,3%	0,0%	-0,1%	103,3%	99,6%
GhostTownFly	0,0%	-0,5%	-0,4%	-0,1%	-0,1%	-0,1%	106,3%	101,8%
PoznanHall2	0,0%	-0,6%	-0,2%	-0,2%	-0,2%	-0,2%	103,7%	99,7%
PoznanStreet	0,0%	-0,4%	-0,2%	-0,2%	-0,1%	-0,1%	104,7%	100,0%
UndoDancer	0,0%	-0,3%	-0,2%	-0,1%	0,0%	0,0%	104,1%	101,4%
1024x768	0,0%	-0,5%	-0,5%	-0,2%	-0,1%	-0,1%	100,6%	100,9%
1920x1088	0,0%	-0,4%	-0,3%	-0,1%	-0,1%	-0,1%	104,7%	100,7%
average	**0,0%**	**-0,4%**	**-0,4%**	**-0,2%**	**-0,1%**	**-0,1%**	**102,9%**	**100,8%**

Furthermore, a simpler version in which the first candidate of the list is systematically selected, whether it is disparity compensated or not, has been implemented. Results are reported in Table 2.

Table 2: Results obtained with HTM 3.0 and Merge list extension with first candidate, compared to HTM 3.0.

	video 0	video 1	video 2	video only	synthesized only	coded & synthesized	enc time	dec time
Balloons	0,0%	-0,3%	-0,4%	-0,2%	#VALEUR!	#VALEUR!	100,8%	101,3%
Kendo	0,0%	-0,4%	-0,5%	-0,2%	#VALEUR!	#VALEUR!	104,2%	102,9%
Newspapercc	0,0%	-0,8%	-0,2%	-0,3%	#VALEUR!	#VALEUR!	104,9%	100,9%
GhostTownFly	0,0%	-0,3%	-0,2%	-0,1%	#VALEUR!	#VALEUR!	108,4%	103,2%
PoznanHall2	0,0%	-0,6%	-0,7%	-0,3%	#VALEUR!	#VALEUR!	106,0%	99,5%
PoznanStreet	0,0%	-0,3%	-0,1%	-0,1%	#VALEUR!	#VALEUR!	107,4%	100,0%
UndoDancer	0,0%	-0,2%	-0,1%	-0,1%	#VALEUR!	#VALEUR!	107,8%	102,2%
1024x768	0,0%	-0,5%	-0,4%	-0,2%	#VALEUR!	#VALEUR!	103,3%	101,7%
1920x1088	0,0%	-0,3%	-0,3%	-0,1%	#VALEUR!	#VALEUR!	107,4%	101,2%
average	**0,0%**	**-0,4%**	**-0,3%**	**-0,2%**	**#VALEUR!**	**#VALEUR!**	**105,6%**	**101,4%**

4 Conclusion

An extension of the merge list is proposed. It is targeted at improving disparity compensated prediction. A 0.4% coding gain on side views on average is obtained. The complexity of the decoder is increased by less than 1%.

Given the reported gains, we propose to include the method in a core experiment for further study.

5 References

[1] Benjamin Bross, Joel Jung, "CE9: Summary Report of Core Experiment on MV Coding and Skip/Merge Operations", JCTVC-G039, JCT-VC 7th Meeting, Geneva, CH, 21-30 November, 2011.

[2] Tomoyuki Yamamoto, Tomohiro Ikai, "Merge candidate refinement for uni-predictive block", JCTVC-I0293, JCT-VC 9th meeting, Geneva, CH, 27 April – 7 May 2012.

[3] Heiko Schwarz, Dmytro Rusanovskyy, « Common test conditions for 3DV experimentation », ISO/IEC JTC1/SC29/WG11 MPEG2011/N12745, May 2012, Geneva, Switzerland.

6 Patent rights declaration(s)

INRIA does not have any current or pending patent rights relating to the technology described in this contribution.