Joint Collaborative Team on Video Coding (JCT-VC)
of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11
8th Meeting: San José, CA, USA, 1–10 February, 2012

Document: JCTVC-H0252

Title: **CE9: Results of SP experiments on simplification of merge process**

Status: Input Document to JCT-VC

Purpose: Proposal

Author(s) or Contact(s): Oguz Bici, Jani Lainema, Kemal Ugur

Email: ext-oguz.bici@nokia.com, jani.lainema@nokia.com, kemal.ugur@nokia.com

Source: Nokia

Abstract

This contribution presents results of CE9 SP experiments related to Part 1 of JCTVC-G593. It aims at reducing complexity of the merge process by proposing to reduce number of motion comparisons during the first pruning process by comparing each spatial candidate with a limited number of spatial candidates and exempting the temporal motion vector prediction (TMVP) candidate from the pruning process. The claimed benefits are decreased complexity by reducing worst case number of duplication checks from 10 to 5, achieving parallel derivation of spatial candidates and the TMVP and increased error resilience by exempting TMVP from duplication check. The reported compression efficiency impact in terms of BD rate is 0.0%, 0.0%, 0.0%, 0.1% and 0.1% for RA_HE, RA_LC, RA_HE_10, LB_HE and LB_LC, respectively.

1 Introduction

Current HM5 design includes a pruning process during the merge mode, where redundant candidates among the initial available spatial and temporal candidates are eliminated from the merge list by comparing motion information of candidates.

During the Geneva Meeting in JCTVC-G593, it was proposed to reduce complexity of the merge list construction by simplifying duplication check processes. In CE9, Part 1 of JCTVC-G593 is further studied and this document presents the details and results of the proposed method. The proposed method is concerned with reducing the number of motion comparisons during the first pruning process by comparing each spatial candidate with a limited number of spatial candidates in a parallel-friendly way and exempting the temporal motion vector prediction candidate from the pruning process.

2 Proposed Method

The description of the experiment is as follows: the first pruning process is removed and instead, it is proposed to perform a smaller number of parallelizable motion information comparisons only between the spatial candidates. It should be noted that the PU availability checks in HM5 (PU's existing, being non-intra, being second PU for the left and above candidates) for the spatial candidates and the number of spatial candidates in HM5 are not modified. The motion information of a spatial candidate is inserted into the merge list depending of a specific condition on this spatial candidate. :

The conditions for each spatial candidate to be excluded from the merge list are as follows:

- A1: No change (which means A1 is always inserted as long as it is available.)

- B1: B1 has the same motion information as A1
- B0: B0 has the same motion information as B1
- A0: A0 has the same motion information as A1
- B2: B2 has the same motion information as A1 or has the same motion information as B1 (Note that as in HM5, B2 is considered only if any of A1, B1, B0 or A0 is excluded from the merge list.)

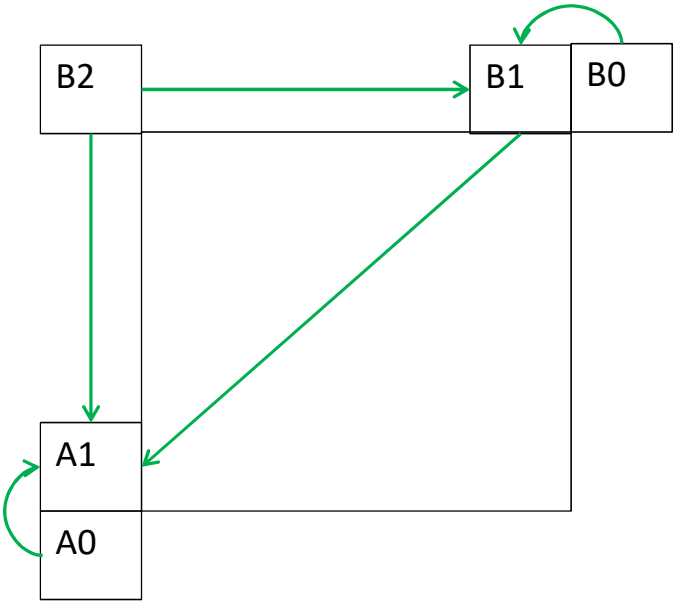

Figure 1. Spatial neighbors which are the potential candidates for the merge list. Arrows indicate which spatial candidate(s) are to be compared.

3 Benefits

- Reduced complexity of merge list generation:
  - Number of worst case duplication checks is reduced to 5 compared to 10 in HM5.
  - All spatial and temporal MVP candidates can be derived independently in parallel. This helps reducing either number of cycles or number of gates in a hardware implementation.
- Decoding time reduction by early exit: With proposed modifications, it becomes possible for a decoder to exit merge process as soon as the candidate corresponding to the decoded merge index is generated. This can be achieved by utilizing the independent candidate removal conditions, which is proposed to replace the process of 'collecting all available candidates first and pruning blindly afterwards'.
- Improved error resilience: This is achieved as the TMVP candidate is not removed from the merge list based on comparing its motion with the previous spatial candidates. In HM5, in case of loss of reference picture(s) and consequently loss of TMVP information, the decoder cannot know if the TMVP is removed from the merge list as it cannot compare its motion information with previous candidates. As a result, it becomes uncertain for a merge index greater than the last spatial candidate to determine if it corresponds to the TMVP or the following additional candidates. This uncertainty is avoided in this step by removing the 1st pruning process.

4 Coding efficiency

HM5.0 software is used for the implementation of the tool and the results are compared against HM5.0 as the anchor under the common test conditions and reference configurations specified in [1]. The results of

the compression efficiency tests are presented in Table 1. Note that encoding times may include some variations due to heterogeneous resources and workloads of the nodes in the simulation cluster. It can be observed that the impact on the compression efficiency is limited.

Table 1

	Random Access HE			Random Access LC			Random Access HE-10		
	Y	U	V	Y	U	V	Y	U	V
Class A (8bit)	0.0%	0.1%	-0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Class B	0.0%	0.0%	0.0%	0.0%	0.2%	0.1%	0.0%	0.0%	0.0%
Class C	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%			
Class D	0.0%	-0.1%	0.0%	0.0%	0.0%	-0.1%			
Class E									
Overall	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%
	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.1%
Class F	-0.1%	0.0%	0.0%	0.1%	0.1%	0.1%			
Enc Time[%]		102%			100%			101%	
Dec Time[%]		97%			100%			99%	

	Low delay B HE			Low delay B LC			Low delay B HE-10		
	Y	U	V	Y	U	V	Y	U	V
Class A									
Class B	0.1%	0.2%	0.5%	0.1%	-0.2%	0.5%			
Class C	0.1%	-0.2%	0.1%	0.1%	0.2%	0.0%			
Class D	0.0%	-0.1%	0.4%	0.0%	0.7%	0.3%			
Class E	0.3%	0.0%	0.3%	0.2%	0.2%	0.5%			
Overall	0.1%	0.0%	0.3%	0.1%	0.2%	0.3%			
	0.1%	0.0%	0.3%	0.1%	0.2%	0.4%			
Class F	-0.1%	0.2%	-0.1%	-0.1%	-0.9%	-0.5%			
Enc Time[%]		100%			100%				
Dec Time[%]		100%			98%				

5 Conclusions

This contribution has presented results of CE9 SP experiments related to Part 1 of JCTVC-G593. As a result of proposed modifications, number of duplication checks during the merge derivation process is reduced to 5 from 10, hardware complexity is reduced by allowing MVP candidates to be derived in parallel and error resiliency is enhanced by exempting TMVP from duplication checks. On the other hand, the impact on the compression efficiency is reported to be limited. The reported compression efficiency impact in terms of BD rate is 0.0%, 0.0%, 0.0%, 0.1% and 0.1% for RA_HE, RA_LC, RA_HE_10, LB_HE and LB_LC, respectively. Therefore, it is recommended to adopt the proposed changes in the next HM and WD.

6 WD Changes

8.4.2.1.1 Derivation process for luma motion vectors for merge mode

This process is only invoked when PredMode is equal to MODE_SKIP or PredMode is equal to MODE_INTER and merge_flag [xP][yP] is equal to 1, where (xP, yP) specify the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture.

Inputs of this process are

– a luma location (xP, yP) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture,

– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH,

– a variable PartIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are

– the luma motion vectors mvL0 and mvL1,

– the reference indices refIdxL0 and refIdxL1,

– the prediction list utilization flags predFlagL0 and predFlagL1.

The motion vectors mvL0 and mvL1, the reference indices refIdxL0 and refIdxL1, and the prediction utilization flags predFlagL0 and predFlagL1 are derived as specified by the following ordered steps:

1. The derivation process for merging candidates from neighboring prediction unit partitions in subclause 8.4.2.1.2 is invoked with luma location (xP, yP), the width and the height of the prediction unit nPSW and nPSH and the partition index PartIdx as inputs and the output is assigned to the availability flags availableFlagN, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N and the motion vectors mvL0N and mvL1N with N being replaced by A_0, A_1, B_0, B_1 or B_2.

2. The reference index for temporal merging candidate is derived as follows..

   – If the prediction unit covering luma location (xP − 1, yP + nPSH − 1) is available and PredMode is not MODE_INTRA, refIdxLX is set equal to refIdxLX[xP − 1, yP + nPSH − 1].

   – Otherwise, refIdxLX is set equal to 0.

3. The derivation process for temporal luma motion vector prediction in subclause 8.4.2.1.9 is invoked with luma location (xP, yP), refIdxLX as the inputs and with the output being the availability flag availableFlagLXCol and the temporal motion vector mvLXCol. The variables availableFlagCol and predFlagLXCol (with X being 0 or 1, respectively) are derived as specified below.

$$availableFlagCol = availableFlagL0Col \,||\, availableFlagL1Col \quad (8\text{-}71)$$

$$predFlagLXCol = availableFlagLXCol \quad (8\text{-}72)$$

4. The merging candidate list, mergeCandList, is constructed as follows.

   1. A_1, if $availableFlagA_1$ is equal to 1
   2. B_1, if $availableFlagB_1$ is equal to 1
   3. B_0, if $availableFlagB_0$ is equal to 1
   4. A_0, if $availableFlagA_0$ is equal to 1
   5. B_2, if $availableFlagB_2$ is equal to 1
   6. Col, if availableFlagCol is equal to 1

5. ~~When merging candidates have the same motion vectors and the same reference indices, the merging candidates are removed from the list except the merging candidate which has the smallest order in the mergeCandList.~~

6. The variable numMergeCand and numOrigMergeCand are set to the number of merging candidates in the mergeCandList.

7. When slice_type is equal to B, the following applies.
   - The derivation process for combined bi-predictive merging candidates specified in subclause 8.4.2.1.4 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N being in mergeCandList, numMergeCand and numOrigMergeCand given as input and the output is assigned to mergeCandList, numMergeCand, the reference indices $refIdxL0combCand_k$ and $refIdxL1combCand_k$, the prediction list utilization flags $predFlagL0combCand_k$ and $predFlagL1combCand_k$ and the motion vectors $mvL0combCand_k$ and $mvL1combCand_k$ of every new candidate $combCand_k$ being added in mergeCandList. The number of candidates being added numCombMergeCand is set equal to (numMergeCand – numOrigMergeCand). When numCombMergeCand is greater than 0, k ranges from 0 to numCombMergeCand − 1, inclusive.
   - The derivation process for non-scaled bi-predictive merging candidates specified in subclause 8.4.2.1.5 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N being in mergeCandList, numMergeCand and numOrigMergeCand given as input and the output is assigned to mergeCandList, numMergeCand, the reference indices $refIdxL0nscaleCand_l$ and $refIdxL1nscaleCand_l$, the prediction list utilization flags $predFlagL0nscaleCand_l$ and $predFlagL1nscaleCand_l$, the motion vectors $mvL0nscaleCand_l$ and $mvL1nscaleCand_l$ of every new candidate $nscaleCand_l$ being added in mergeCandList. The number of candidates being added numNscaleMergeCand is set equal to (numMergeCand – numOrigMergeCand – numCombMergeCand). When numNscaleMergeCand is greater than 0, l ranges from 0 to numNscaleMergeCand – 1, inclusive.
8. The derivation process for zero motion vector merging candidates specified in subclause 8.4.2.1.6 is invoked with the mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N being in mergeCandList and the NumMergeCand as the inputs and the output is assigned to mergeCandList, numMergeCand, the reference indices $refIdxL0zeroCand_m$ and $refIdxL1zeroCand_m$, the prediction list utilization flags $predFlagL0zeroCand_m$ and $predFlagL1zeroCand_m$, the motion vectors $mvL0zeroCand_m$ and $mvL1zeroCand_m$ of every new candidate $zeroCand_m$ being added in mergeCandList. The number of candidates being added numZeroMergeCand is set equal to (numMergeCand – numOrigMergeCand – numCombMergeCand – numNscaleMergeCand). When numZeroMergeCand is greater than 0, m ranges from 0 to numZeroMergeCand – 1, inclusive.
9. The following assignments are made with N being the candidate at position merge_idx[xP][yP] in the merging candidate list mergeCandList (N = mergeCandList[merge_idx[xP][yP]]) and X being replaced by 0 or 1:

$$mvLX[\ 0\ ] = mvLXN[\ 0\ ] \tag{8-88}$$

$$mvLX[\ 1\ ] = mvLXN[\ 1\ ] \tag{8-89}$$

$$refIdxLX = refIdxLXN \tag{8-90}$$

$$predFlagLX = predFlagLXN \tag{8-91}$$

8.4.2.1.2 Derivation process for spatial merging candidates

Inputs to this process are

– a luma location (xP, yP) specifying the top-left luma sample of the current prediction unit relative to the top-left sample of the current picture,

– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH,

– a variable PartIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are (with N being replaced by A_0, A_1, B_0, B_1 or B_2 and with X being replaced by 0 or 1)

– the availability flags availableFlagN of the neighbouring prediction units,

– the reference indices refIdxLXN of the neighbouring prediction units,

– the prediction list utilization flags predFlagLXN of the neighbouring prediction units,

– the motion vectors mvLXN of the neighbouring prediction units.

For the derivation of availableFlagN, with N being A_0, A_1, B_0, B_1 or B_2 and (xN, yN) being (xP – 1, yP + nPSH), (xP − 1, yP + nPSH − 1), (xP + nPSW, yP – 1), (xP + nPSW − 1, yP – 1) or (xP – 1, yP – 1), the following applies.

– If yP-1 is smaller than YLCU, the following applies.

xB_0 = (xB_0>>Log2MinCUSize)<<Log2MinCUSize) + ((xB_0>>Log2MinCUSize)&1)*(MinPuSize*2-1) (8-88)
xB_1 = (xB_1>>Log2MinCUSize)<<Log2MinCUSize) + ((xB_1>>Log2MinCUSize)&1)*(MinPuSize*2-1) (8-88)
xB_2 = (xB_2>>Log2MinCUSize)<<Log2MinCUSize) + ((xB_2>>Log2MinCUSize)&1)*(MinPuSize*2-1) (8-88)

where MinPuSize = 1 << (Log2MinCUSize – 1)

– If one of the following conditions is true with X being replaced by 0 and 1, the availableFlagN is set equal to 0, both components mvLXN are set equal to 0, refIdxLXN and predFlagLX[xN, yN] of the prediction unit covering luma location (xN, yN) are assigned respectively to mvLXN, refIdxLXN and predFlagLXN.

  – N is equal to B_2 and availableFlagA_0 + availableFlagA_1 + availableFlagB_0 + availableFlagB_1 is equal to 4.

  – The prediction unit covering luma location (xN, yN) is not available or PredMode is MODE_INTRA.

  – PartMode of the current prediction unit is PART_2NxN or PART_2NxnU or PART_2NxnD and PartIdx is equal to 1 and N is equal to B1

  – PartMode of the current prediction unit is PART_Nx2N or PART_nLx2N or PART_nRx2N and PartIdx is equal to 1 and N is equal to A1

  – N is equal to B1 and the prediction units covering luma location (xP − 1, yP + nPSH − 1) (N = A1) and luma location (xN, yN) (Cand. N) have the same motion vectors and the same reference indices

  – N is equal to B0 and the prediction units covering luma location (xP + nPSW − 1, yP – 1) (N = B1) and luma location (xN, yN) (Cand. N) have the same motion vectors and the same reference indices

  – N is equal to A0 and the prediction units covering luma location (xP − 1, yP + nPSH − 1) (N = A1) and luma location (xN, yN) (Cand. N) have the same motion vectors and the same reference indices

  – N is equal to B2 and the prediction units covering luma location (xP + nPSW − 1, yP – 1) (N = B1) and luma location (xN, yN) (Cand. N) have the same motion vectors and the same reference indices

– N is equal to B2 and the prediction units covering luma location (xP − 1, yP + nPSH − 1) (N = A1) and luma location (xN, yN) (Cand. N) have the same motion vectors and the same reference indices

– Otherwise, availableFlagN is set equal to 1 and the variables mvLX[xN, yN], refIdxLX[xN, yN] and predFlagLX[xN, yN] of the prediction unit covering luma location (xN, yN) are assigned respectively to mvLXN, refIdxLXN and predFlagLXN.

7 References

[1] F. Bossen, "Common test conditions and software reference configurations," JCTVC-G1200, Nov 2011

8 Patent rights declaration(s)

Nokia 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).