Joint Collaborative Team on Video Coding (JCT-VC)
of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11
7th Meeting: Geneva, CH, 21-30 November, 2011

Document: JCTVC-G593

Title: **Non-CE13: Simplification of merge mode**

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 comprises three parts aiming at reducing complexity of the merge process. In the first part, it is proposed 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 candidate from the pruning process. In the second part, it is proposed to derive redundancy information for the combined candidates during the first pruning process and exempting other additional candidates from pruning. In the third part, number of contexts used for the merge index is reduced to two and the contexts are assigned according to being a skip mode or inter-merge mode. Combination of all the three parts provides the merge mode having a total of one full motion comparison per PU where other motion comparisons are either removed or replaced by simpler operations. The reported compression efficiency impact in terms of BD rate is 0.1%, 0.1%, 0.1%, 0.2% for RA_HE, LB_HE, RA_LC and LB_LC, respectively.

1 Introduction

Current HM4 design includes two pruning processes during the merge mode. In the first pruning process, redundant candidates among the initial available spatial and temporal candidates are eliminated from the merge list by comparing motion information of candidates. In the second pruning process, redundant candidates among the additional candidates, which are generated to obtain a merge list of 5 candidates, are excluded from the merge list by comparing their motion information to previously added merge candidates. This contribution comprises three parts aiming at reducing complexity of the merge process associated with candidate generation and motion information comparison:

- Part 1: This part 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 and exempting the temporal motion vector prediction candidate from the pruning process.
- Part 2: This part is concerned with decreasing the complexity of the second pruning process by deriving redundancy information for the combined candidates during the first pruning process and exempting other additional candidates (non-scaled and zero candidates) from redundancy checks. In this way, no motion comparison is needed during the second pruning process.
- Part 3: In this part, number of contexts used for coding the merge index with CABAC is reduced to two and the contexts are assigned according to being a skip mode or inter-merge mode.

Nokia Exhibit 2016, p. 1
Amazon.com v. Nokia
IPR2024-00604

2 Part 1: Simplification of the first pruning process

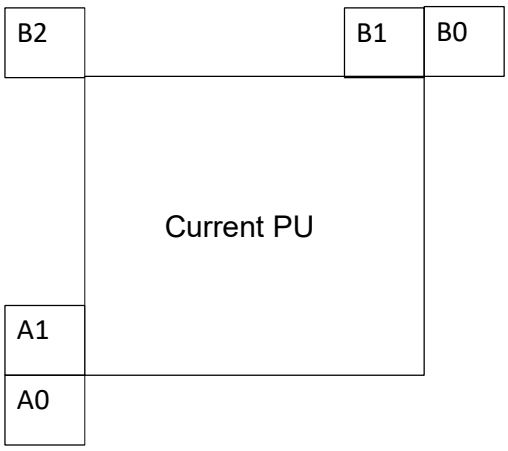

Figure 1 Five spatial locations used during the merge process

In the current HM4, the merge process initially considers five spatial candidates (located at A1, B1, B0, A0 and B2 in Figure 1) and a temporal candidate. A spatial candidate is identified as unavailable if the associated block does not exist or is intra or the current PU is a 2nd PU of a rectangle CU and the spatial candidate has same motion as the 1st PU. Then, maximum of 4 available spatial candidates and the temporal candidate (if available) undergo the 1st pruning process where redundant candidates are removed from the merge list. After the removals, the remaining candidates constitute the initial MVP list.

Part 1 of this contribution aims to reduce complexity of generation of the initial candidates, which correspond to the generation of candidates with the 1st pruning process in HM4.. For this purpose, it is proposed to perform a smaller number of motion comparisons between the candidates and to use simpler comparison logic for some of the candidates. The proposed changes are as follows:

1. The condition of "the current PU being a 2nd PU of a rectangle CU and the spatial candidate having same motion as the 1st PU", called as the "2nd PU condition" throughout the contribution, for determining availability of each spatial candidate is modified. The modified conditions to identify if the spatial candidate is unavailable for each spatial candidate are as follows:
    - A1: No change with the 2nd PU condition
        - Note that, rather than blindly comparing motion of A1 with that of 1st PU, equivalently, it is proposed to check whether the current CU is vertically or horizontally splitted into two PUs and decide accordingly. Because if it is vertical, then the 1st PU already covers A1 and there is no need to perform a motion comparison. If it is horizontal, then the 1st PU has same motion as B1. Therefore, it is sufficient to compare motions of A1 and B1 and the result of this comparison is also shared with the availability derivation for B1, as seen below.
    - B1: In addition to the 2nd PU condition, following condition is added:
        - B1 has same motion as A1
        - Note that, again similarly for the 2nd PU condition, rather than blindly comparing motion of B1 with that of 1st PU, it is sufficient to check if the CU is horizontally split into two PUs. Because in this case, the 1st PU already covers B1 and there is no need to perform motion comparison. Moreover, there is no need to check if the CU is vertically split since in this case, the 1st PU would cover A1 and motion comparison of B1 and A1 is always checked for B1 in the proposed method.
    - B0: The 2nd PU condition is removed, following condition is added:
        - B0 has same motion as B1
    - A0: The 2nd PU condition is removed, following condition is added:

Nokia Exhibit 2016, p. 2
Amazon.com v. Nokia
IPR2024-00604

        - A0 has same motion as A1
    - B2: The 2nd PU condition is removed, following condition is added:
        - B2 has same motion as A1 or has same motion as B1

2. In order to reduce complexity of comparing the motion information of the candidate pairs B0 with B1, A0 with A1, B2 with A1 and B2 with B1, it is proposed to define and process simpler variables. This is achieved by defining for each 4x4 partition (or simply partition) the following 2 types of data: the information of how many consecutive partitions to the left have the same motion information and the information of how many consecutive partitions to the above have the same motion information. Updating these values is described later in this Section. Utilizing these two types of data, the motion comparisons during the availability check for B0, A0 and B2 are realized as follows:
    - B0 has same motion as B1: The number of consecutive partitions with same motion to the left of B0 is greater than 0
    - A0 has same motion as A1: The number of consecutive partitions with same motion to the above of A0 is greater than 0
    - B2 has same motion as B1: The number of consecutive partitions with same motion to the left of B1 is greater than the number of partitions between B2 and B1
    - B2 has same motion as A1: The number of consecutive partitions with same motion to the above of A1 is greater than the number of partitions between B2 and A1
3. The 1st pruning process is removed, i.e. when the available spatial and temporal candidates are obtained, no more motion comparison between the candidates is performed. In addition to the complexity reduction, this step has also side benefit of improved error resiliency. This is achieved as the temporal motion vector prediction candidate is not removed from the merge list based on comparing its motion with the previous spatial candidates. In HM4, 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.

As mentioned, it is proposed to define for each 4x4 partition the information of number of consecutive partitions on the left/above with same motions. Each of them (left and above) require 5 bits (total of 10 bits) per 4x4 partition as the maximum number of consecutive partitions checked is 16 for a PU with a maximum dimension of 64. These data are updated after the motion information of an inter PU is decoded. It is enough to update/store only bottom-most partitions of the PU for the information of number of consecutive partitions on the left with same motions; and only right-most partitions of the PU for the information of number of consecutive partitions above with same motions. If the current PU is a non-merge inter PU, it is assumed that left and above neighbor PUs have different motion and number of partitions with same motion on the left/above information is updated only within the current PU as illustrated in Figure 2. Otherwise, if the current PU is using a merge mode (inter merge or skip CU), the consecutively same motion information of left (A1 partition) and above (B1 partition) neighbors is utilized during the update. This is achived by making use of:

- the information whether A1 and B1 has same motion, which is already obtained during the merge process
- the information whether the selected merge candidate is A1 or B1

For updating the right-most partitions, first it is determined whether the PU above has same motion by checking if the selected merge candidate is B1, or the selected merge candidate is A1 and A1 has same motion as B1. Then, the right-most partitions are updated by adding the value of number of consecutive partitions having same motion above for the partition located at B1. For updating the bottom-most partitions, first it is determined whether the PU on the left has same motion by checking if the selected

Nokia Exhibit 2016, p. 3
Amazon.com v. Nokia
IPR2024-00604

merge candidate is A1, or the selected merge candidate is B1 and A1 has same motion as B1. Then, the bottom-most partitions are updated by adding the value of number of consecutive partitions having same motion on the left for the partition located at A1. The procedure is depicted in Figure 3.

Note that it is not required to update all the partitions of the PU; it is sufficient to update only bottom-most and right-most partitions. Moreover, the update procedure can be carried in parallel with the subsequent processes taking place after the motion vector is obtained.

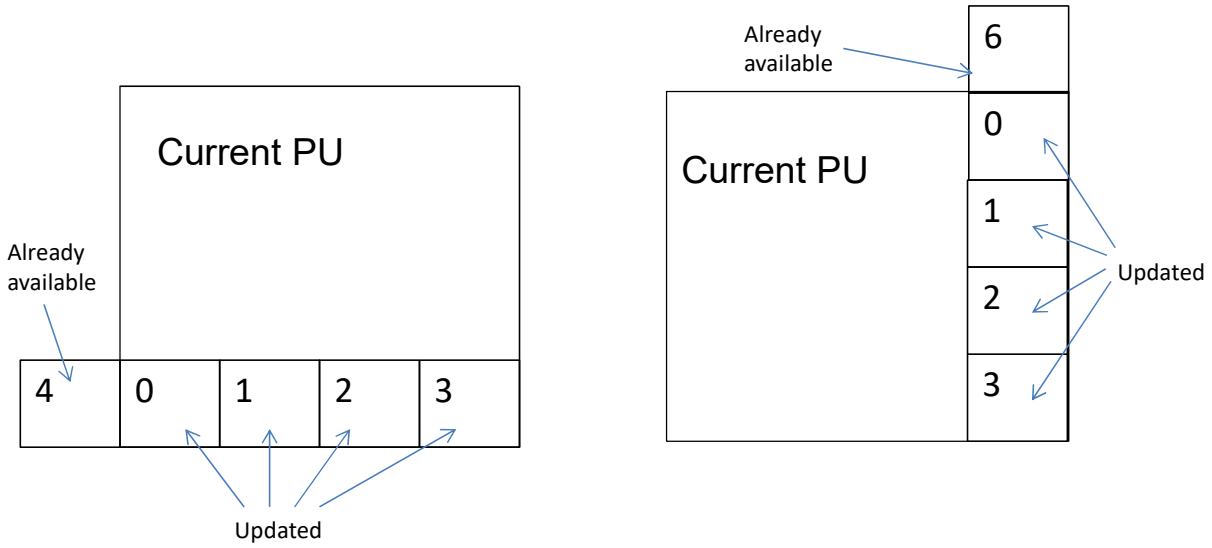

Figure 2 Updating consecutively same motion information for a non-merge inter PU. Left side: Updating number of partitions with same motions on the left for bottom-most partitions. Right side: Updating number of partitions with same motions above for right-most partitions.

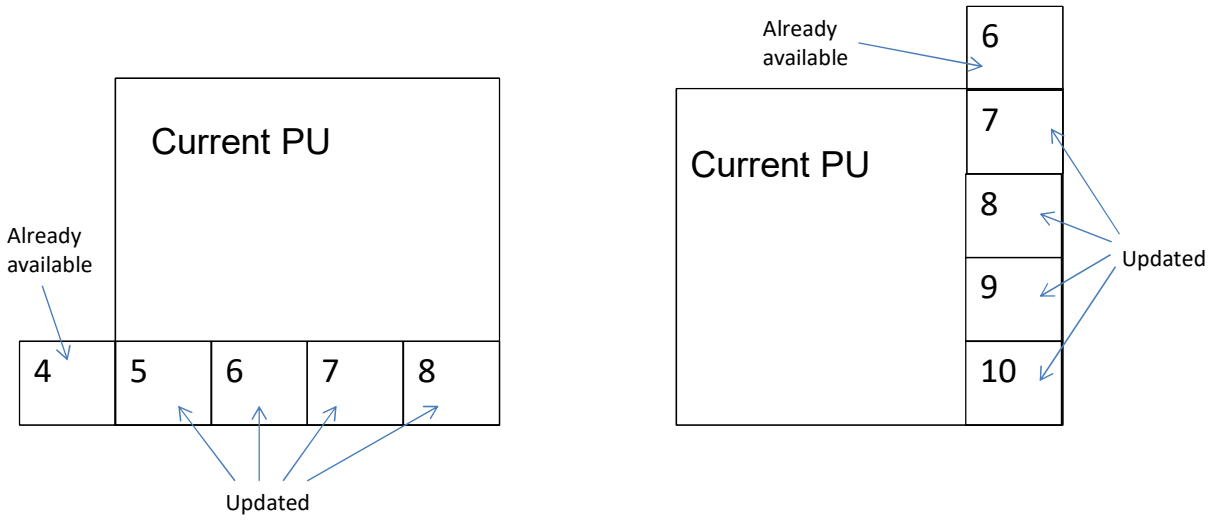

Figure 3 Updating consecutively same motion information for a merge based PU. Left side: Updating number of partitions with same motions on the left for bottom-most partitions. Right side: Updating number of partitions with same motions above for right-most partitions.

2.1 Experimental results

HM4.0 software is used for the implementation of the tool and the results are compared against HM4.0 as the anchor under the common test conditions and reference configurations specified in [1]. The same common test conditions and reference configurations are used in the rest of the contribution. Compression efficiency performance of Part 1 is presented in Table 1.

Table 1 Results of Part 1

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.1%	-0.1%	0.0%	0.0%	0.1%	0.0%
Class B	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%
Class C	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%
Class D	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E						
Overall	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]		N/A			N/A	
Dec Time[%]		95%			99%	

	Low delay B HE			Low delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.1%	0.0%	0.1%	0.1%	0.2%	0.3%
Class C	0.1%	-0.1%	0.1%	0.1%	0.1%	0.0%
Class D	0.1%	0.2%	0.3%	0.1%	0.2%	-0.3%
Class E	0.2%	0.7%	1.0%	0.2%	0.7%	0.3%
Overall	0.1%	0.1%	0.3%	0.1%	0.3%	0.1%
	0.1%	0.1%	0.3%	0.1%	0.2%	0.0%
Enc Time[%]		N/A			N/A	
Dec Time[%]		96%			95%	

2.2 Comments

With the proposed changes described in Part 1, it is possible obtain the initial MVP list by performing a single motion comparison whereas HM4 requires 10 motion comparisons during the first pruning process in the worst case. In the proposed method, some of the motion comparisons are removed and some of them are replaced by simpler operations. It is observed that the impact on the compression efficiency is limited. In addition, the proposed changes are more suitable for parallel processing, and reduces memory bandwidth requirements since motion and reference index information of some of the candidates are not required. Also, with the removal of redundancy checks of the first pruning process, it becomes possible for a decoder to exit merge process as soon as the candidate corresponding to the decoded merge index is generated. Finally, as a side benefit, error resiliency of the merge process is improved since the temporal motion vector predictor is not removed from the list based on comparing its motion information.

3 Part 2: Simplification of the second pruning process

In this part, the goal is to simplify the 2nd pruning process during the merge mode. In HM4, after the 1st pruning process, additional candidates (combined, non-scaled and zero candidates) are generated and added to the merge list if its motion information is not same as a previously added candidate. In this contribution, it is proposed to completely remove the redundancy checks of the additional candidates.

Nokia Exhibit 2016, p. 5
Amazon.com v. Nokia
IPR2024-00604

This is achieved by deriving redundancy information for specific pairs of the combined candidates during the 1st pruning process (to be described in detail), and removing redundancy checks for the non-scaled and zero additional candidates.

Derivation of redundancy information for the combined candidates: During the 1st pruning process, redundant candidates are eliminated by comparing the motion information with other candidates. The comparison includes the whole motion information, i.e. comparing equivalence of both List0 and List1. Assume that two candidates have the same List0 motion but different List1 motion. In this case, none of the candidates are eliminated during the 1st pruning process. However, it can be derived that combinations of these candidates will be redundant. Moreover, combination of the second candidate with remaining candidates will be redundant as well. Similar arguments are also valid in case List1 motions are the same but List0 motions are different.

The aforementioned redundancies are illustrated with an example in Figure 4. In the example, there are three candidates after the 1st pruning (Cand 0,1,2) and four example combinations of these candidates (Comb Cand 0&1, 1&0, 0&2, 1&2). Motion information of the candidates per list is denoted by MotX, X={A,B,C,D,E}. In the example, Cand 0 and Cand 1 have the same List0 motion. As a result, as seen in the Figure, combination of Cand 0 and Cand 1 has same motion as Cand 1 and combination of Cand 1and Cand 0 has same motion as Cand 0, making these combined candidates redundant. Moreover, it is also observed that combination of Cand 0 and Cand 2 has same motion as combination of Cand 1and cand 2, making the latter combination redundant.

Candidates	List0	List1
Cand 0	MotA	MotB
Cand 1	MotA	MotC
Cand 2	MotD	MotE
Comb Cand 0&1	MotA	MotC
Comb Cand 1&0	MotA	MotB
Comb Cand 0&2	MotA	MotE
Comb Cand 1&2	MotA	MotE

Figure 4 Example case of three candidates from the initial MVP list and four combinations of these candidates

In order to utilize these pre-detectable redundancies, it is proposed to define an availability map for the combined candidates, which is initialized by setting all the combinations as available. Then this combined availability mapis updated during the 1st pruning process. Whenever two candidates are being compared during 1st pruning process, if either list 0 or list 1 motion information is identified as same, then redundant combined candidates are derived and updated as unavailable in the availability map. Figure 5 shows an example candidate list before the combined candidates are generated, Figure 6 shows a corresponding initialized availability map, and Figure 7 and Figure 8 show example update cases when motion information of candidates 0 and 1 are same for the lists 0 and 1, respectively.

	Reference Picture list0		Reference Picture list1	
CandIdx	MV	RefIdx	MV	RefIdx
0	MV0List0	RefIdx0List0	MV0List1	RefIdx0List1
1	MV1List0	RefIdx1List0	MV1List1	RefIdx1List1
2	MV2List0	RefIdx2List0	MV2List1	RefIdx2List1
3	MV3List0	RefIdx3List0	MV3List1	RefIdx3List1

Figure 5 Example candidate list

Nokia Exhibit 2016, p. 6
Amazon.com v. Nokia
IPR2024-00604

CandIdx	0	1	2	3
0	0	1	1	1
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0

Figure 6 Initialized combined availability map

CandIdx	0	1	2	3
0	0	0	1	1
1	0	0	0	0
2	1	1	0	1
3	1	1	1	0

Figure 7 updating the combined availability map when MV0List0 is equal to MV1List0 and RefIdx0List0 is equal to RefIdx1List0

CandIdx	0	1	2	3
0	0	0	1	1
1	0	0	1	1
2	1	0	0	1
3	1	0	1	0

Figure 8 updating the combined availability map when MV0List1 is equal to MV1List1 and RefIdx0List1 is equal to RefIdx1List1

As a result, at the end of the 1st pruning, there exists an availability map for the combined candidates, describing which combinations are redundant. In this way, it is proposed to remove redundancy check of the combined candidates with the previous candidates; instead, it is proposed to use the derived availability map to determine whether it is redundant. For the remaining non-scaled and zero candidates, it is proposed to disable redundancy checks without further modifications.

3.1 Experimental results

HM4.0 software is used for the implementation of the tool and the results are compared against HM4.0 as the anchor. Table 2 shows the compression efficiency performance of only using the combined availability map, where the redundancy check for the non-scaled and zero candidates is still applicable. Table 3 shows the compression efficiency performance of Part 2, where combined availability map is used for the combined candidates as well as the redundancy check for the non-scaled and zero candidates is disabled.

Table 2 Results of using only the combined availability map

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.0%	-0.2%	-0.1%	0.0%	0.2%	0.0%
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E						
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%
Enc Time[%]		N/A			N/A	

Nokia Exhibit 2016, p. 7
Amazon.com v. Nokia
IPR2024-00604

Dec Time[%]	100%	100%

	Low delay B HE			Low delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	-0.2%	-0.1%	0.0%	0.0%	0.0%
Class C	0.0%	-0.1%	-0.1%	0.0%	0.1%	-0.1%
Class D	0.0%	0.1%	0.1%	0.0%	0.6%	0.0%
Class E	0.0%	0.2%	0.5%	0.0%	0.4%	-0.1%
Overall	0.0%	0.0%	0.1%	0.0%	0.2%	0.0%
	0.0%	0.0%	0.0%	0.0%	0.2%	-0.1%
Enc Time[%]	N/A			N/A		
Dec Time[%]	101%			100%		

Table 3 Results of Part 2 (both using combined availability map and disabling redundancy check for the non-scaled and zero candidates)

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.1%	0.0%	-0.1%	0.0%	0.0%	0.1%
Class B	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%
Class C	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%
Class D	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%
Class E						
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%
	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%
Enc Time[%]	N/A			N/A		
Dec Time[%]	100%			100%		

	Low delay B HE			Low delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class C	0.0%	-0.1%	0.0%	0.1%	0.2%	-0.1%
Class D	0.0%	0.3%	-0.1%	0.0%	-0.1%	-0.1%
Class E	0.1%	0.0%	0.8%	0.1%	0.4%	0.4%
Overall	0.0%	0.1%	0.1%	0.1%	0.1%	0.0%
	0.0%	0.1%	0.1%	0.1%	0.1%	0.0%
Enc Time[%]	N/A			N/A		
Dec Time[%]	96%			100%		

3.2 Comments

In summary, Part 2 of this contribution provides a realization of the 2nd pruning process which does not include any motion comparisons. Compared to the worst case of HM4, HM4 requires 60 motion comparisons during the second pruning process [3]. This is achieved by utilizing partial list motion equivalence information, which can be obtained without extra effort during the 1st pruning. Experimental results show that the proposed complexity reduction is achieved with almost no effect on the compression performance.

Nokia Exhibit 2016, p. 8
Amazon.com v. Nokia
IPR2024-00604

4 Combination of Part 1 and Part 2

In this part, the ideas presented in Part 1 and Part 2 are jointly utilized. In order to generate the availability map of Part 2, the simplified set of motion comparisons presented in Part 1 is used. Since partial list motion comparisons are required, the information of number of consecutive motions with same motions on the left/above is now stored for each list.

In summary, combination of Part 1 and Part 2 results in the merge mode having only one full motion comparison per PU. Other comparisons for redundancy removals are either removed or replaced by simpler operations.

4.1 Experimental Results

Compression efficiency results for the combination of Part 1 and Part 2 are presented in Table 4 and Table 5. In Table 4, the anchor is HM4.0. In Table 5, the fix related to merge mode selection in the encoder released by Samsung on CE13 reflector is applied to both the proposed combination of Part 1 and 2 and the anchor. This fix is denoted by MrgEncFix throughout the rest of this document.

Table 4 Results of combination of Part 1 and 2; anchor is HM4

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.1%	-0.1%	-0.1%	0.1%	0.1%	0.0%
Class B	0.1%	0.1%	0.0%	0.1%	0.1%	0.1%
Class C	0.1%	0.1%	0.1%	0.1%	0.0%	0.1%
Class D	0.1%	0.1%	0.0%	0.1%	0.0%	0.0%
Class E						
Overall	0.1%	0.0%	0.0%	0.1%	0.1%	0.0%
	0.1%	0.0%	0.0%	0.1%	0.1%	0.1%
Enc Time[%]		N/A			N/A	
Dec Time[%]		99%			99%	

	Low delay B HE			Low delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.2%	0.1%	0.1%	0.2%	0.2%	0.2%
Class C	0.1%	0.1%	0.2%	0.1%	0.2%	0.0%
Class D	0.2%	0.4%	0.6%	0.2%	0.2%	0.2%
Class E	0.3%	0.7%	1.4%	0.6%	1.2%	0.9%
Overall	0.2%	0.3%	0.5%	0.2%	0.4%	0.3%
	0.2%	0.3%	0.5%	0.2%	0.4%	0.3%
Enc Time[%]		N/A			N/A	
Dec Time[%]		97%			98%	

Table 5 Results of combination of Part 1 and 2 with MrgEncFix; anchor is HM4 with MrgEncFix

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.1%	0.1%	0.1%	0.1%	0.0%	-0.1%
Class B	0.1%	0.2%	0.2%	0.1%	0.1%	0.0%
Class C	0.1%	0.0%	0.1%	0.1%	0.1%	0.0%
Class D	0.1%	0.1%	0.2%	0.1%	-0.1%	0.0%
Class E						
Overall	0.1%	0.1%	0.2%	0.1%	0.0%	0.0%
	0.1%	0.1%	0.2%	0.1%	0.0%	0.0%

Nokia Exhibit 2016, p. 9
Amazon.com v. Nokia
IPR2024-00604

Enc Time[%]	N/A	N/A
Dec Time[%]	99%	99%

	Low delay B HE			Low delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.2%	0.2%	0.1%	0.2%	0.1%	0.0%
Class C	0.1%	0.0%	0.1%	0.2%	0.3%	0.2%
Class D	0.1%	0.4%	0.4%	0.1%	-0.3%	0.2%
Class E	0.4%	0.3%	0.2%	0.4%	0.4%	0.5%
Overall	0.2%	0.2%	0.2%	0.2%	0.1%	0.2%
	0.2%	0.3%	0.2%	0.2%	0.1%	0.2%
Enc Time[%]	N/A			N/A		
Dec Time[%]	97%			97%		

4.2 Comments

It is observed that impact of the combination of Part1 and Part2 on the compression efficiency is still limited while the complexity reductions described previously are retained.

5 Part 3: Modified context assignments for the merge index

Part 3 of this contribution concerns with the coding of the merge index with CABAC. Currently in HM4, four contexts are used, one per each bin, for the merge index. In this part, it is proposed to use a total of two contexts, both assigned to coding of the first bin. One of the contexts is used when the merge mode is used for a Skip CU and the other is used when the merge mode is a Merge PU. The remaining three bins are coded with equal probability.

6 Experimental results for combination of Part 1, Part 2 and Part 3

Compression efficiency results for the combination of all proposed parts (Part 1 + Part 2 + Part 3) are presented in Table 6 and Table 7. In Table 6, the anchor is HM4.0. In Table 7, the MrgEncFix is applied to both the proposed combination and the anchor HM4.0.

Table 6 Results of combination of all proposed parts; anchor is HM4

	Random Access HE		
	Y	U	V
Class A	0.1%	-0.1%	-0.2%
Class B	0.1%	0.1%	0.1%
Class C	0.1%	0.0%	0.2%
Class D	0.1%	0.2%	0.0%
Class E			
Overall	0.1%	0.1%	0.0%
	0.1%	0.1%	0.1%
Enc Time[%]	N/A		
Dec Time[%]	99%		

	Low delay B HE		
	Y	U	V
Class A			
Class B	0.1%	0.0%	0.0%
Class C	0.1%	0.1%	0.0%
Class D	0.1%	0.6%	-0.2%
Class E	0.3%	0.8%	1.4%
Overall	0.1%	0.3%	0.2%

Nokia Exhibit 2016, p. 10
Amazon.com v. Nokia
IPR2024-00604

	0.1%	0.3%	0.2%
Enc Time[%]		N/A	
Dec Time[%]		97%	

Table 7 Results of combination of all proposed parts with MrgEncFix; anchor is HM4 with MrgEncFix

	Random Access HE		
	Y	U	V
Class A	0.1%	0.1%	-0.1%
Class B	0.1%	0.2%	0.2%
Class C	0.1%	0.1%	0.2%
Class D	0.1%	0.1%	0.2%
Class E			
Overall	0.1%	0.1%	0.1%
	0.1%	0.1%	0.2%
Enc Time[%]		N/A	
Dec Time[%]		99%	

	Low delay B HE		
	Y	U	V
Class A			
Class B	0.1%	0.4%	0.0%
Class C	0.1%	-0.1%	0.2%
Class D	0.1%	0.6%	0.3%
Class E	0.3%	0.5%	0.1%
Overall	0.1%	0.3%	0.1%
	0.1%	0.3%	0.1%
Enc Time[%]		N/A	
Dec Time[%]		99%	

6.1 Comments

It can be noticed that the application of Part3 improves the compression efficiency of Part+Part2 by 0.1% BD-rate for Low Delay B configuration and achieves similar efficiency for Random Access configuration. Moreover, number of contexts for coding the merge index is reduced to two from four and three out of four bins of the merge index are coded with equal probability.

7 Conclusion

In this contribution, three parts aiming at reducing the complexity of the merge candidate generation have been presented. With the application of all parts in the contribution, the worst case number of required motion comparisons per merge PU in the decoder is reduced to 1 from 70 (10 in the first pruning and 60 in the second pruning). Also, application of the proposed modifications may lead to simpler logic and/or increased throughput and/or improved average memory bandwidth. On the other hand, the impact on the compression efficiency is reported to be limited. The resultant compression performance in terms of BD rate is reported to be 0.1%, 0.1%, 0.1%, 0.2% for RA_HE, LB_HE, RA_LC and LB_LC, respectively. Therefore, it is recommended to adopt the proposed changes in the next HM and WD.

8 References

[1] F. Bossen, "Common test conditions and software reference configurations," JCTVC-F900, Jul 2011

[2] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, T. Wiegand, “WD4: Working Draft 4 of High-Efficiency Video Coding,” JCTVC-F803, Jul 2011

Nokia Exhibit 2016, p. 11
Amazon.com v. Nokia
IPR2024-00604

[3] M. Zhou, "CE13: Test results on maxNumMergeCand signaling and simplification of merge MVP list pruning process," JCTVC-G091, Nov 2011

9 WD Changes

Changes to WD are provided as separate documents.

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).

Nokia Exhibit 2016, p. 12
Amazon.com v. Nokia
IPR2024-00604