



CE8-related: Default filling method for IBC mode

Shurui Ye, **Yucheng Sun**, Fangdong Chen, Li Wang

2019年7月1日

- In the current design of IBC, if the number of available IBC merge candidates is less than 6, default zero block vectors are added until the merge list is fully filled.
- However, zero block vectors are **invalid** for IBC mode.
- This contribution presents a default filling method for IBC mode.

- **Method 1: *Filling the merge list with $(-2*cbWidth, 0)$ block vector***

If the merge list is still not complete, the default block vector of $(-2*cbWidth, 0)$ is used to fill the merge list, where, $cbWidth$ is the coding block width in luma samples, and block vector in integer pixel accuracy.

5. When $numCurrCand$ is less than $MaxNumMergeCand$, the following applies until $numCurrCand$ is equal to $MaxNumMergeCand$:
 - $mvCandList[numCurrCand][0]$ is set equal to $(-2 * cbWidth << 4)$.
 - $mvCandList[numCurrCand][1]$ is set equal to 0.
 - $numCurrCand$ is increased by 1.

Technical description

- **Method 2: *Filling the merge list with a default block vector list***

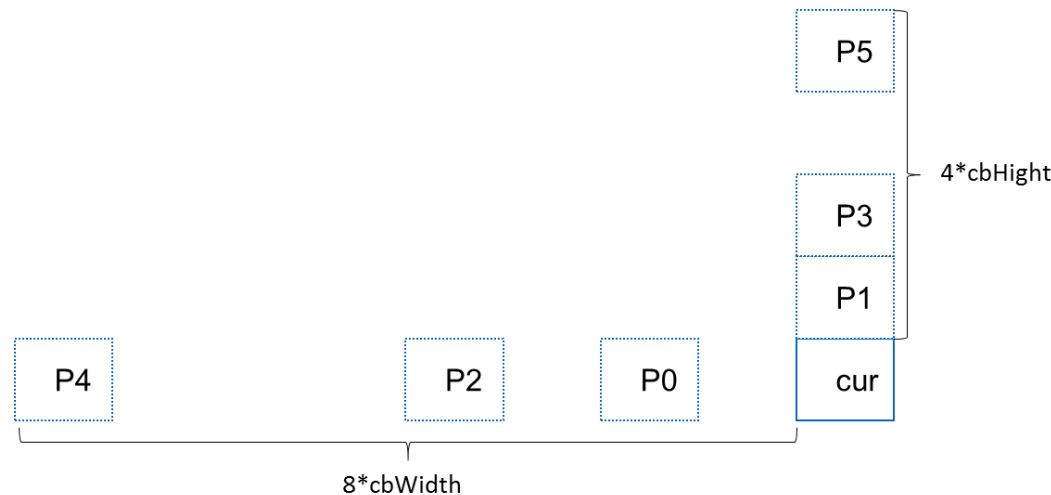
The default block vectors are added in the order of:

$(-2*cbWidth, 0), (0, -1*cbHeight),$

$(-4*cbWidth, 0), (0, -2*cbHeight),$

$(-8*cbWidth, 0), (0, -4*cbHeight),$

where $cbWidth$ and $cbHeight$ are the coding block width and height in luma samples, respectively.



- **Method 2: *Filling the merge list with a default block vector list***

5. When numCurrCand is less than MaxNumMergeCand, for each candidate in DefaultCandList[DefaultIdx] with index = 1...MaxNumMergeCand, the following applies until numCurrCand is equal to MaxNumMergeCand:
- mvCandList[numCurrCand][0] is set equal to $\ominus$ DefaultCandList[DefaultIdx++][0].
 - mvCandList[numCurrCand][1] is set equal to $\ominus$ DefaultCandList[DefaultIdx++][1].
 - numCurrCand is increased by 1.

•Table 8-15 –Default candidates

DefaultIdx	DefaultCandList[DefaultIdx][0]	DefaultCandList[DefaultIdx][1]
1	$-2 * cbWidth \ll 4$	0
2	0	$-1 * cbHight \ll 4$
3	$-4 * cbWidth \ll 4$	0
4	0	$-2 * cbHight \ll 4$
5	$-8 * cbWidth \ll 4$	0
6	0	$-4 * cbHight \ll 4$

Simulation results

Thanks Bytedance and Alibaba for crosschecking

- **Method 1: *Filling the merge list with $(-2*cbWidth, 0)$ block vector***

	Over VTM5.0 (AI configuration)				
	Y	U	V	EncT	DecT
Class F	-0.04%	0.03%	-0.02%	103%	100%
TGM	-0.05%	-0.03%	-0.03%	101%	101%

	Over VTM5.0 (RA configuration)				
	Y	U	V	EncT	DecT
Class F	-0.09%	-0.18%	0.11%	106%	100%
TGM	-0.06%	-0.03%	-0.08%	103%	100%

*Common test conditions for Class F are followed.

Simulation results

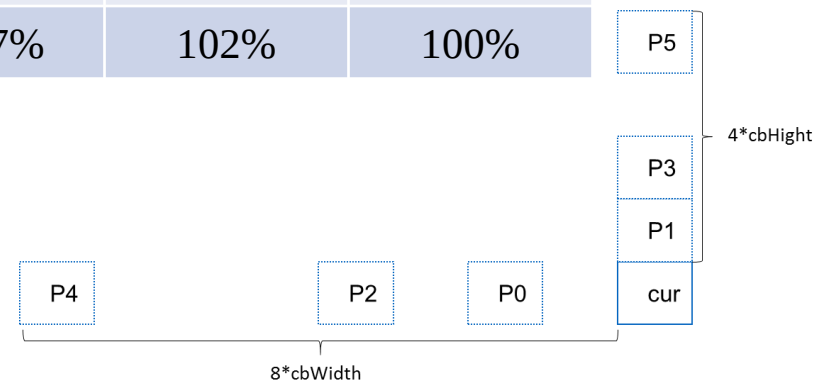
Thanks Bytedance and Alibaba for crosschecking

- **Method 2: Filling the merge list with a default block vector list**

	Over VTM5.0 (AI configuration)				
	Y	U	V	EncT	DecT
Class F	-0.07%	0.05%	-0.03%	102%	101%
TGM	-0.09%	-0.05%	-0.05%	101%	100%

	Over VTM5.0 (RA configuration)				
	Y	U	V	EncT	DecT
Class F	-0.13%	-0.16%	-0.13%	104%	100%
TGM	-0.16%	-0.21%	-0.27%	102%	100%

*Common test conditions for Class F are followed.



- Proposed default filling method for IBC mode
 - Method 1: Filling the merge list with $(-2*cbWidth, 0)$ block vector
 - AI: 0.05% gain for TGM
 - RA: 0.06% gain for TGM
 - Method 2: Filling the merge list with a default block vector list
 - AI: 0.09% gain for TGM
 - RA: 0.16% gain for TGM
- We suggest to adopt the proposed default filling method to VVC.



Thanks

科技呵护未来

— First Choice for Security Professionals —