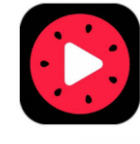


JVET-N0263

CE2-5.5: HISTORY-PARAMETER-BASED AFFINE MODEL INHERITANCE

Kai Zhang, Li Zhang, Hongbin Liu, Jizheng Xu,
Yue Wang, Pengwei Zhao, Dingkun Hong
Bytedance Inc.

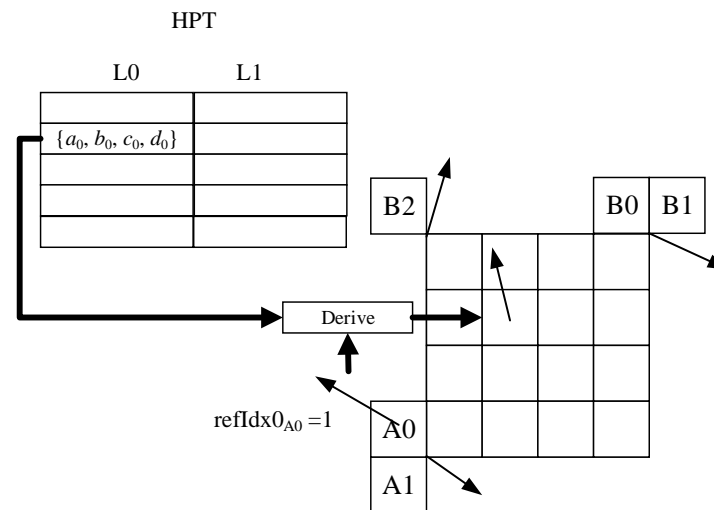


History-Parameter Table

- A History-Parameter Table (HPT) is established with 10 entries
 - *Each entry stores a set of affine parameters: a , b , c and d , each of which is represented by an 8-bit signed integer.*
 - *The total size of HPT is $10 \times 4 \times 8 = 320$ bits.*
 - *The entry index is derived as: $HPTIdx(RefList, RefIdx) = 5 \times RefList + \min(RefIdx, 4)$*
- At the beginning of each CTU row, all entries in HPT are initialized as zero.
- After decoding an affine-coded CU with reference list $RefList_{cur}$ and $RefIdx_{cur}$, the affine parameters are utilized to rewrite $HPT[HPTIdx(RefList_{cur}, RefIdx_{cur})]$, if not all of parameters are equal to 0.

History-Parameter-based Affine Candidate

- A History-Parameter-based Affine Candidate (HPAC) is derived from a neighbouring 4×4 block
- An example:
 - The affine parameters $\{a_0, b_0, c_0, d_0\}$ are directly copied from $HPT[HPTIdx(RefList_{A0}, refIdx0_{A0})]$.
 - If $\{a_0, b_0, c_0, d_0\} == \{0, 0, 0, 0\}$, HPAC is unavailable for block A1.
 - Otherwise, the affine parameters from HPAC, with the center position of A0 as the base position, and the MV of block A0 as the base MV, are used together to derive the MV of each sub-block in the current CU for a merge HPAC, or to derive the CPMVs for an AMVP HPAC.



Sub tests

- In VTM-4.0, the first affine coded left neighbouring 4×4 block and the first affine coded above neighbouring 4×4 block are found from the five neighbouring 4×4 blocks (A0, A1, B0, B1 and B2). Then the affine inheritance candidates are derived from these two neighbouring blocks and put in the candidate list as the first two affine candidates.
- In the proposed method, affine candidates derived from these two neighbouring 4×4 blocks are kept as the first two candidates.
 - *In CE2-5.5.a, they are the original affine inheritance candidates.*
 - *In CE2-5.5.b, they are HPACs.*
 - *In CE2-5.5.c, the candidate from the above neighbouring block may be the original affine inheritance candidate if it is in the above CTU-row.*
- *HPACs from other inter-coded neighbouring blocks are put behind.*

Comparison on storage

	In-CTU buffer Size	Line buffer size
VTM-4.0	7689 bits	8 bits/8×8
CE2-5.5.a	8009 bits (104% as anchor)	8 bits/8×8 (100% as anchor)
CE2-5.5.b	320 bits (4% as anchor)	0 (0% as anchor)
CE2-5.5.c	320 bits (4% as anchor)	8 bits/8×8 (100% as anchor)

Simulations results on VTM-4.0

■ Test a:

	RA					LB				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
Class A1	-0.06%	0.08%	-0.11%	102%	101%					
Class A2	-0.14%	-0.20%	-0.09%	101%	100%					
Class B	-0.06%	-0.07%	0.05%	101%	101%	-0.01%	-0.34%	-0.28%	99%	99%
Class C	-0.02%	-0.09%	-0.22%	101%	100%	-0.07%	0.04%	0.06%	101%	102%
Class E						0.05%	-0.01%	-0.10%	101%	103%
Overall	-0.06%	-0.07%	-0.08%	101%	101%	-0.02%	-0.13%	-0.12%	100%	101%
Class D	-0.05%	-0.17%	-0.15%	100%	103%	-0.17%	-0.24%	-0.38%	101%	102%
Class F	-0.04%	0.06%	-0.06%	100%	102%	-0.14%	0.20%	0.25%	100%	96%

■ Test b:

	RA					LB				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
Class A1	-0.01%	0.03%	-0.03%	101%	99%					
Class A2	0.03%	-0.03%	0.01%	100%	99%					
Class B	-0.01%	-0.07%	-0.05%	99%	101%	0.08%	-0.28%	-0.16%	100%	99%
Class C	0.00%	-0.06%	-0.02%	99%	101%	0.03%	0.26%	0.05%	101%	101%
Class E						0.22%	0.40%	-0.07%	100%	95%
Overall	0.00%	-0.04%	-0.02%	100%	100%	0.10%	0.07%	-0.07%	100%	99%
Class D	0.01%	0.08%	0.00%	99%	100%	0.01%	0.00%	-0.67%	100%	103%
Class F	0.05%	0.04%	-0.10%	99%	103%	0.00%	-0.33%	-0.04%	98%	100%

Simulations results on VTM-4.0

■ Test c:

	RA					LB				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
Class A1	-0.07%	-0.11%	-0.03%	101%	100%					
Class A2	-0.06%	-0.10%	-0.09%	100%	99%					
Class B	-0.06%	-0.14%	-0.17%	100%	100%	0.03%	-0.15%	-0.27%	99%	98%
Class C	-0.01%	-0.09%	-0.15%	99%	100%	-0.02%	0.18%	-0.03%	99%	101%
Class E						0.04%	-0.08%	-0.04%	101%	97%
Overall	-0.05%	-0.11%	-0.12%	100%	100%	0.02%	-0.02%	-0.13%	99%	99%
Class D	0.01%	0.01%	-0.11%	99%	105%	-0.03%	0.48%	-0.46%	101%	105%
Class F	0.03%	0.05%	-0.06%	99%	103%	-0.09%	0.23%	0.31%	98%	99%

Conclusion

- Considering the good trade-off between memory requirement and coding performance, it is recommended to adopt CE2-5.5.c into VVC.