JVET-Y0161

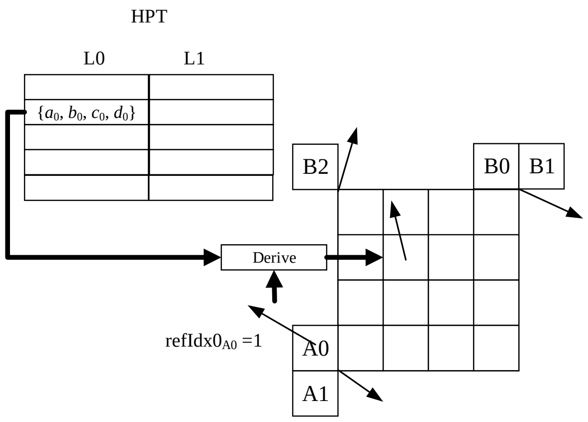
EE2-3.12-RELATED: EXTENSIONS OF HISTORY-PARAMETER-BASED AFFINE MODEL INHERITANCE

Kai Zhang, Li Zhang, Zhipin Deng, Na Zhang, Yang Wang

Bytedance Inc.

Introduction

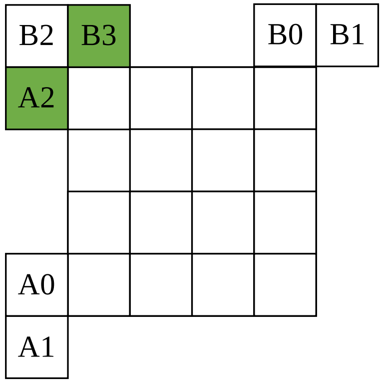
- History-parameter-based affine model inheritance (HAMI) in JVET-X0088 is tested in EE2-3.12a
- A History-Parameter Table (HPT) is established with 70 entries
  - *Each entry stores a set of affine parameters: {a, b, c, d}*
  - *Entries in HPT is categorized by reference list and reference index*
    - HPTCat (RefList, RefIdx)= 5 × RefList + min(RefIdx, 4)
    - Seven entries in each category
  - *HPT is refreshed at the beginning of each CTU row*
  - *HPT is updated after decoding an affine coded block*



- A history-affine-parameter-based candidate (HAPC) is jointly generated from a neighbouring 4×4 block (such as A0) and an entry in HPT
  - *$\{ a_0, b_0, c_0, d_0 \}$ are fetched from one entry of category $HPTCat(RefList_{A0}, refIdx0_{A0})$*
  - *Affine parameters from HPAC, with the center position of A0 as the base position (x_{base}, y_{base}), and the MV of block A0 as the base MV $(mv^h_{base}, mv^v_{base})$, are used together to derive CPMVs*

$$\begin{cases} mv^h(x,y) = a(x - x_{base}) + c(y - y_{base}) + mv^h_{base} \\ mv^v(x,y) = b(x - x_{base}) + d(y - y_{base}) + mv^v_{base} \end{cases}$$

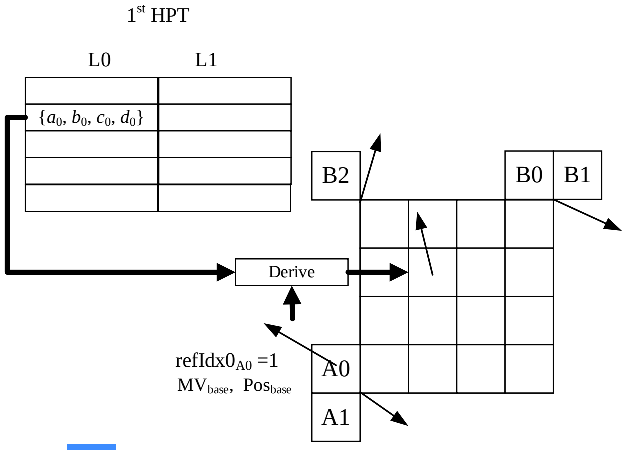
Extensions of EE2-3.12a

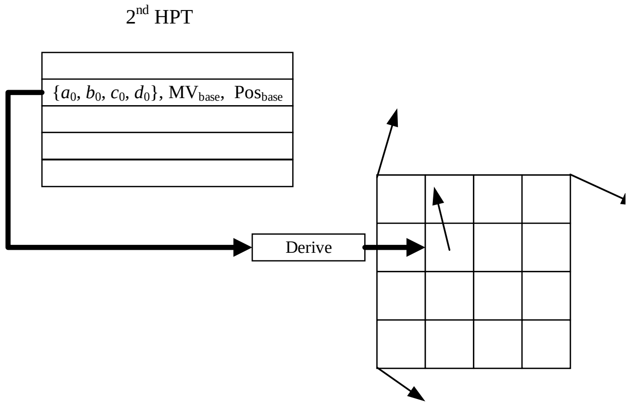


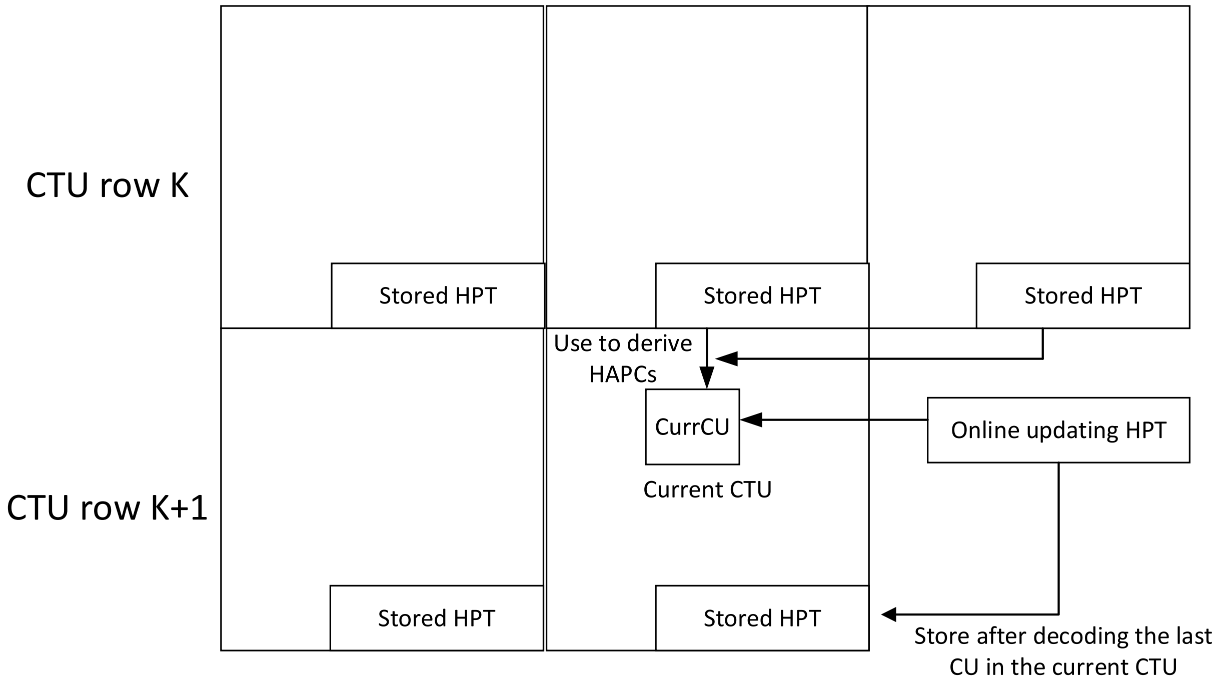
- HAMI is implemented strictly following the EE description which may limit the performance
- Based on the code of EE2-3.12, some extensions are proposed
  - *Aspect #1: The size of sub-block-based merge candidate list is increased to 15, and all the affine merge candidates are involved in the adaptive reordering merge candidates (ARMC) process*
  - *Aspect #2: The number of neighboring blocks, which are used to generate affine merge candidates jointly with affine models stored in tables, are increased from 5 to 7*
  - *Aspect #3: A second history-parameter table (HPT) with base MV information is appended. And the history-parameter tables stored in the CTU above/above-right to the current CTU*
    - Nine entries in the second HPT. Each entry comprises affine parameters and base MV/base position
    - An additional HAPC can be generated with the base MV information and the affine models stored in an entry
  - *Aspect #4: Pair-wised affine merge candidates are generated by two affine merge candidates which are history-derived or not history-derived*
    - A pair-wised affine merge candidates is generated by averaging the CPMVs of two existing candidates

Illustration of Aspect #3

- Difference between the 1st HPT and the 2nd HPT
  - *1st HPT only comprises affine models. A HAPC is jointly generated from base MV/base position of a neighbouring 4×4 block and an entry in 1st HPT*
  - *2nd HPT comprises both affine models and base MV/base position. One entry in 2nd HPT can generate a HAPC independently*
- After coding/decoding a CTU, the HPTs will be stored for the usage in the next CTU row







Simulations results on ECM-3.1

Test #1: EE2-3.12 + Aspect #1

	RA					LB				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
Class A1	-0.25%	-0.28%	-0.28%	102%	102%					
Class A2	-0.43%	-0.23%	-0.31%	102%	103%					
Class B	-0.24%	-0.15%	-0.07%	100%	101%	-0.34%	-0.29%	-0.10%	100%	100%
Class C	-0.16%	-0.12%	-0.03%	98%	97%	-0.23%	0.09%	-0.06%	100%	100%
Class E						-0.14%	0.13%	-0.44%	100%	99%
Overall	**-0.26%**	**-0.19%**	**-0.15%**	**100%**	**100%**	**-0.25%**	**-0.06%**	**-0.17%**	**100%**	**100%**
Class D	-0.17%	0.03%	-0.04%	100%	96%	-0.32%	0.23%	-0.14%	101%	100%
Class F	-0.11%	-0.10%	-0.11%	99%	97%	-0.31%	-0.30%	-0.12%	100%	96%

Test #2: EE2-3.12 + Aspect #1~#4

	RA					LB				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
Class A1	-0.31%	-0.38%	-0.32%	99%	100%					
Class A2	-0.54%	-0.40%	-0.36%	99%	101%					
Class B	-0.33%	-0.27%	-0.19%	100%	101%	-0.42%	-0.30%	-0.44%	99%	102%
Class C	-0.20%	-0.01%	-0.14%	98%	99%	-0.27%	-0.37%	-0.06%	99%	101%
Class E						-0.19%	0.04%	-0.44%	99%	100%
Overall	**-0.34%**	**-0.25%**	**-0.24%**	**99%**	**100%**	**-0.31%**	**-0.24%**	**-0.31%**	**99%**	**101%**
Class D	-0.27%	-0.04%	-0.05%	99%	99%	-0.53%	0.11%	-0.87%	97%	99%
Class F	-0.22%	-0.20%	-0.28%	99%	100%	-0.52%	-0.22%	-0.23%	96%	98%

Conclusion

- Extensions on EE of HAMI are proposed
  - *0.3%+ RA coding gain in average*
  - *Negligible complexity increase*
- Recommend to be adopted in ECM
- Thanks to KDDI for Crosschecking!