

G776: Merge candidates list construction

Tammy Lee, Jianle Chen, Jeonghoon Park
Samsung Electronics Co., Ltd

Nov. XX, 2011

Introduction

- ❖ In the current HM4.0, merge index is signaled assuming the maximum number of candidates as 5
 - If some of pre-defined merge candidates are duplicated or unavailable, the number of candidates cannot reach to 5 and it improves performance to insert additional merge candidates in such cases
 - The more candidates increase complexity while the more candidates improve coding efficiency

- ❖ Proposed merge list construction
 - Includes encoder side bug fix which affects performance when additional candidates are inserted
 - Includes additional candidates which are inserted only if the pre-defined merge candidates are less than the maximum number of candidates
 - Includes the way to provide trade-off between complexity and coding efficiency

Encoder side bugfix

- ❖ In HM4.0, the number of bits for merge index coding is not considered during merge estimation stage for normal merge modes
 - Merge candidates having larger merge index and higher rd cost may be inserted without consideration of cost for merge index coding
 - This is critical if the more candidates are inserted
- ❖ It is proposed that the number of bits for merge index coding is considered as estimated bits without entropy coding in merge estimation stage

	Y	U	V	Enc	Dec
HE_RA	-0.1%	-0.1%	-0.2%	100%	101%
LC_RA	-0.1%	-0.1%	0.0%	100%	100%
HE_LB	-0.2%	-0.2%	0.0%	100%	99%
LC_LB	-0.1%	0.0%	-0.2%	100%	101%
HE_LD	-0.1%	-0.1%	-0.2%	100%	100%
LC_LD	0.0%	0.0%	-0.1%	100%	100%

Additional candidates (1)

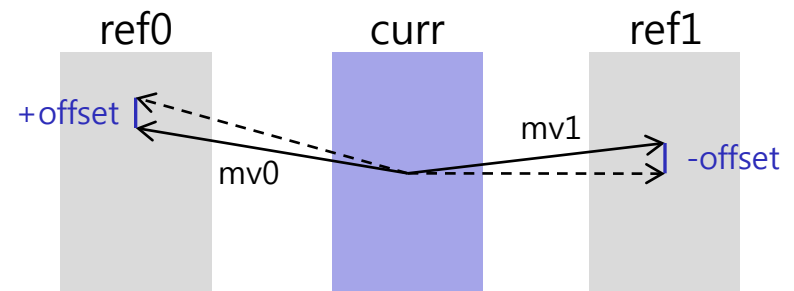
- ❖ To avoid fetching motion information from new positions, the existing first candidate is modified to generate alternatives
- ❖ In this test, pre-defined zero merge candidates are reduced to 1
- ❖ Offset added candidates
 - If the number of pre-defined merge candidates is less than the maximum candidates' list size, offset added candidates inserted
 - InterDir and reference index(indices) of the existing first candidate are copied into offset added candidates
 - If the existing first candidate is uni-directional, the offset is added to the one MV
 - Otherwise (if it is bi-directional), the offset is added to both MVs

Additional candidates (2)

❖ Offset based candidates (Cont'd)

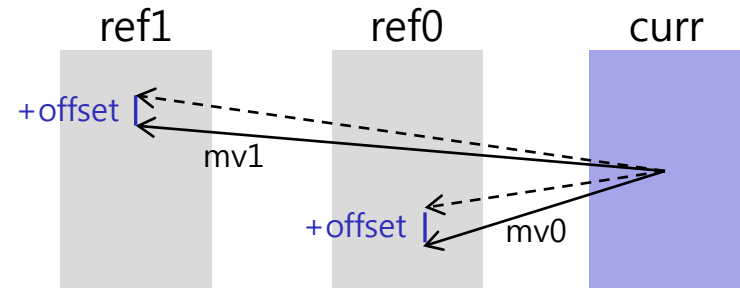
- Offsets are added as follows:
(for an example of $mv0=(mx_0, my_0)$ and $mv1=(mx_1, my_1)$ of the first candidate)
- If the current picture lies between two reference pictures

- (mx_0+1, my_0) and (mx_1-1, my_1)
- (mx_0-1, my_0) and (mx_1+1, my_1)
- (mx_0, my_0+1) and (mx_1, my_1-1)
- (mx_0, my_0-1) and (mx_1, my_1+1)



- if the current picture lies in the same direction from two reference pictures

- (mx_0+1, my_0) and (mx_1+1, my_1)
- (mx_0-1, my_0) and (mx_1-1, my_1)
- (mx_0, my_0+1) and (mx_1, my_1+1)
- (mx_0, my_0-1) and (mx_1, my_1-1)



Additional candidates (3)

❖ Experimental result

- Additional candidates are tested on top of 'HM4.0+Encoder bugfix' and it compared to 'HM4.0+Encoder bugfix' as anchor

	Y	U	V	Enc	Dec
HE_RA	-0.1%	0.0%	0.0%	102%	100%
LC_RA	-0.2%	0.0%	-0.1%	102%	99%
HE_LB	-0.1%	0.3%	0.0%	102%	101%
LC_LB	-0.1%	0.4%	0.1%	102%	100%
HE_LD	-0.2%	0.2%	0.2%	104%	100%
LC_LD	-0.4%	0.2%	-0.2%	104%	101%

- In average, -0.2% gain with 102% encoder runtime increment
- Encoder runtime is increased because more interpolation process to check RD cost for more candidates are needed in estimation

Adaptive merge list size (1)

❖ Slice based merge list size signal

- It is proposed to signal the maximum merge list size `maxNumMergeCand` of each slice in slice header
- `maxNumMergeCand` can vary from 1 to 5
- In this test, `maxNumMergeCand` of each slice is derived according to merge list candidates statistics of previous coded slices
 - 1) For each slice, the average of the number of the first step candidates which are derived before combined candidates is calculated
 - 2) This average value keeps being updated with the previous average value as an average of previous average value and the newest average value for each temporal layer
 - 3) This average value is used as the maximum merge list size after being added by the offset value which is determined on temporal layer or slice type

Adaptive merge list size (2)

❖ Experimental result

	Y	U	V	Enc	Dec
HE_RA	0.1%	0.1%	0.1%	97%	100%
LC_RA	0.1%	0.1%	0.1%	97%	101%
HE_LB	0.2%	0.2%	0.2%	98%	100%
LC_LB	0.2%	0.2%	0.1%	98%	101%
HE_LD	-0.1%	0.1%	-0.1%	96%	100%
LC_LD	-0.1%	-0.2%	-0.2%	95%	100%

- By restriction on the maximum list size, encoder complexity can be reduced
- Gain even can be achieved in such LD cases that overhead bits are burdened

Conclusion

- ❖ In this proposal, encoder side bug fix of HM4.0, additional merge candidates and high level syntax to signal max merge list size are introduced
- ❖ Encoder side bug fix which considers merge index bits in estimation stage improves performance without complexity increment
- ❖ Slice header syntax to signal max merge list size gives the flexibility to control the trade-off of complexity and performance. This can be set as the default value 5 in common test configuration without performance change
- ❖ Additional merge candidates by adding offsets to the existing candidate can improve performance. With optimal offset setting other than 1, this performance can be further improved
- ❖ It is proposed to have new slice header and new offset added merge candidates with optimal value setting of maximum merge list size and offset value

Thank you !