AHG8: Modification of History Based Rice Parameter Derivation

JVET-X0127

Yue Yu, Haoping Yu, Zhihuang Xie, Fan Wang, Luhang Xu, Dong Wang

History-based Rice parameter derivation

- A history-based Rice parameter derivation method JVET-V0106 was adopted for VVC V2;
- A HistValue is calculated to update locSumAbs to derive new Rice parameter;
- A counter/color, StatCoeff[3] is utilized and may be updated once per TU from the first abs_remainder[] or dec_abs_level[]

StatCoeff[i] = (StatCoeff[i] +Floor(Log2(abs_remainder[]) + 2) >> 1

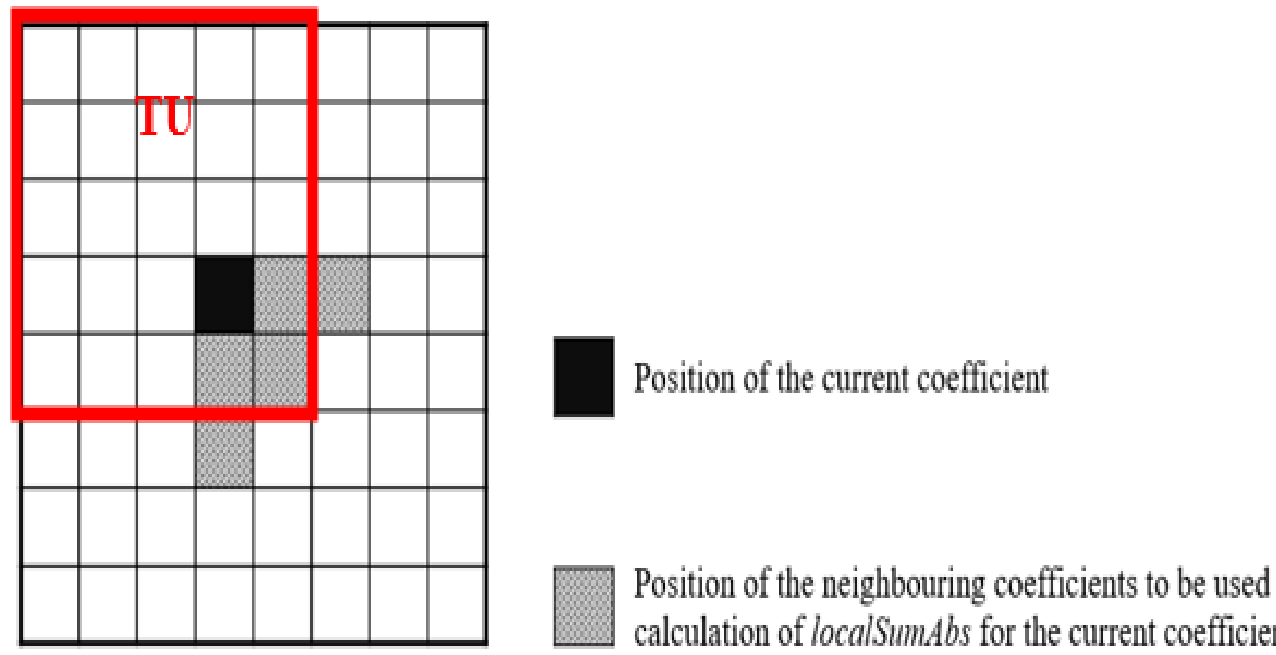
or

StatCoeff[i] = (StatCoeff[i] +Floor(Log2(abs_remainder[])) >> 1

Prior to TU coding, HistValue = 1 << StatCoeff[i]

And StatCoeff[i] is initialed as 2* Floor(Log2(BitDepth - 10))

- The updated StatCoeff will be used to derive a new HistValue for coding the next TU.




```
locSumAbs = 0
if( xC < ( 1  <<  log2TbWidth ) − 1 ) {
  locSumAbs  +=  AbsLevel[ xC + 1 ][ yC ]
  if( xC < ( 1  <<  log2TbWidth ) − 2 )
    locSumAbs  +=  AbsLevel[ xC + 2 ][ yC ]
  else
    locSumAbs  +=  HistValue
  if( yC < ( 1  <<  log2TbHeight ) − 1 )
    locSumAbs  +=  AbsLevel[ xC + 1 ][ yC + 1 ]
  else
    locSumAbs  +=  HistValue
} else
  locSumAbs  +=  2 * HistValue
if( yC < ( 1  <<  log2TbHeight ) − 1 ) {
  locSumAbs  +=  AbsLevel[ xC ][ yC + 1 ]
  if( yC < ( 1  <<  log2TbHeight ) − 2 )
    locSumAbs  +=  AbsLevel[ xC ][ yC + 2 ]
  else
    locSumAbs  +=  HistValue
} else
  locSumAbs  +=  HistValue
```

Residual coding

…	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight − (log2SbW + log2SbH))) − 1	
HistValue = sps_persistent_rice_adaptation_enabled_flag ? (1 << StatCoeff[cIdx]) : 0	
updateHist = sps_persistent_rice_adaptation_enabled_flag ? 1 : 0	
…	
if(abs_level_gtx_flag[n][1]) {	
abs_remainder[n]	ae(v)
if(updateHist && abs_remainder[n] > 0) {	
StatCoeff[cIdx] = (StatCoeff[cIdx] + Floor(Log2(abs_remainder[n])) + 2) >> 1	
updateHist = 0	
}	
}	
…	
if(sb_coded_flag[xS][yS]) {	
dec_abs_level[n]	ae(v)
if(updateHist && dec_abs_level[n] > 0) {	
StatCoeff[cIdx] = (StatCoeff[cIdx] + Floor(Log2(dec_abs_level[n]))) >> 1	
updateHist = 0	
}	
}	

Proposed Method

- Once the StatCoeff is updated, the updated value is used to derive a new HistValue and it will be used immediately for coding all the remaining abs_remainder[] and dec_abs_level[] within the current TU.
-

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	**Descriptor**
…	
}	
for(n = firstPosMode0; n > firstPosMode1; n− −) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(abs_level_gtx_flag[n][1]) {	
abs_remainder[n]	ae(v)
if(updateHist && abs_remainder[n] > 0) {	
StatCoeff[cIdx] = (StatCoeff[cIdx] + Floor(Log2(abs_remainder[n])) + 2) >> 1	
updateHist = 0	
HistValue = 1 << StatCoeff[cIdx]	
}	
}	
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] +2 * abs_remainder[n]	
}	
for(n = firstPosMode1; n >= 0; n− −) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(sb_coded_flag[xS][yS]) {	
dec_abs_level[n]	ae(v)
if(updateHist && dec_abs_level[n] > 0) {	
StatCoeff[cIdx] = (StatCoeff[cIdx] + Floor(Log2(dec_abs_level[n]))) >> 1	
updateHist = 0	
HistValue = 1 << StatCoeff[cIdx]	
}	
}	
if(AbsLevel[xC][yC] > 0) {	
if(lastSigScanPosSb = = −1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
if(sh_dep_quant_used_flag)	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
}	
…….	

Simulation Results

- Simulation conditions follow the CTC for high bit-depth coding;
- Over VTM-14.0, low QP, lossy.

HDR PQ	AI							
	Over VTM14.0							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	#VALUE!	#VALUE!	#VALUE!	-0.01%	0.00%	-0.02%	100%	101%
PQ422	#VALUE!	#VALUE!	#VALUE!	-0.01%	-0.02%	-0.01%	99%	99%
Overall	#VALUE!	#VALUE!	#VALUE!	-0.01%	-0.01%	-0.02%	100%	100%
	LDB							
	Over VTM14.0							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	#VALUE!	#VALUE!	#VALUE!	0.00%	-0.04%	-0.01%	100%	100%
PQ422	#VALUE!	#VALUE!	#VALUE!	0.01%	0.01%	0.03%	100%	99%
Overall	#VALUE!	#VALUE!	#VALUE!	0.00%	-0.02%	0.01%	100%	100%
	RA							
	Over VTM14.0							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	-0.01%	0.00%	0.01%	-0.01%	0.00%	0.01%	99%	100%
PQ422	0.00%	-0.01%	-0.02%	0.00%	-0.01%	-0.02%	97%	100%
Overall	-0.01%	-0.01%	0.00%	-0.01%	-0.01%	-0.01%	98%	100%
Overall PQ	#VALUE!	#VALUE!	#VALUE!	0.00%	-0.01%	0.00%	99%	100%

HDR HLG	AI				
	Over VTM14.0				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	-0.01%	-0.02%	-0.01%	98%	100%
HLG422	-0.02%	-0.01%	-0.01%	99%	100%
Overall	-0.02%	-0.01%	-0.01%	99%	100%
	LDB				
	Over VTM14.0				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	0.00%	0.00%	0.01%	100%	100%
HLG422	0.00%	-0.03%	0.02%	101%	100%
Overall	0.00%	-0.01%	0.01%	101%	100%
	RA				
	Over VTM14.0				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	0.01%	0.00%	0.00%	99%	100%
HLG422	0.00%	0.01%	0.00%	101%	100%
Overall	0.00%	0.00%	0.00%	100%	100%
Overall HLG	-0.01%	-0.01%	0.00%	100%	100%

SVT RGB	AI				
	Over VTM14.0				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	-0.02%	-0.03%	-0.03%	100%	100%
SVT12	-0.03%	-0.03%	-0.03%	100%	100%
Overall	-0.03%	-0.03%	-0.03%	100%	100%
	LDB				
	Over VTM14.0				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	-0.03%	-0.03%	-0.03%	101%	100%
SVT12	-0.01%	-0.01%	-0.01%	101%	100%
Overall	-0.02%	-0.02%	-0.02%	101%	100%
	RA				
	Over VTM14.0				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	-0.04%	-0.03%	-0.03%	101%	101%
SVT12	-0.01%	-0.01%	-0.01%	101%	100%
Overall	-0.02%	-0.02%	-0.02%	101%	101%
Overall RGB	-0.02%	-0.02%	-0.02%	101%	100%

- Over VTM-14.0, lossless

PQ	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
PQ444	2.6	2.6	-0.03%	3.1	3.1	-0.04%	3.1	3.1	-0.04%
PQ422	2.4	2.4	-0.03%	2.9	2.9	-0.04%	2.9	2.9	-0.04%
Overall	**2.5**	**2.5**	**-0.03%**	**3.0**	**3.0**	**-0.04%**	**3.0**	**3.0**	**-0.04%**
Enc Time[%]	103%			93%			91%		
Dec Time[%]	96%			95%			92%		

HLG	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
HLG444	1.8	1.8	-0.01%	2.0	2.0	-0.01%	2.0	2.0	-0.01%
HLG422	1.7	1.7	-0.02%	1.9	1.9	-0.02%	1.9	1.9	-0.02%
Overall	**1.7**	**1.7**	**-0.02%**	**2.0**	**2.0**	**-0.02%**	**2.0**	**2.0**	**-0.02%**
Enc Time[%]	103%			93%			91%		
Dec Time[%]	99%			99%			93%		

SVT	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
SVT16	1.2	1.2	-0.02%	1.2	1.2	-0.01%	1.2	1.2	-0.02%
SVT12	1.3	1.3	-0.02%	1.3	1.3	-0.03%	1.3	1.3	-0.03%
Overall	**1.2**	**1.2**	**-0.02%**	**1.3**	**1.3**	**-0.02%**	**1.3**	**1.3**	**-0.02%**
Enc Time[%]	104%			100%			100%		
Dec Time[%]	101%			92%			104%		

Conclusion

- A simple modification allows the updated StatCoeff to be used to derive HistValue immediately, and this new HistValue value is used to compute Rice parameter for all the remaining coefficients within the current TU.
- Change to the VVC-v2 specification is very simple and straightforward.
- Coding gains are achieved for both lossy and lossless conditions.

Thank Qualcomm for crosschecking!

Thank you

OPPO