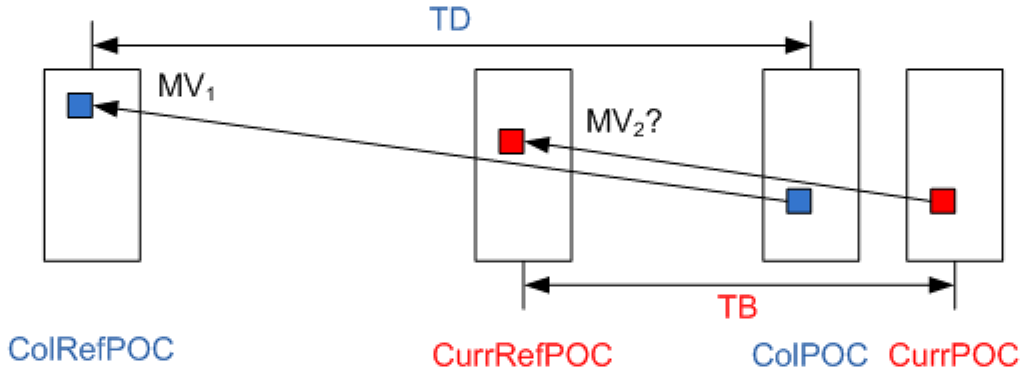
JCTVC-J0155: SIMPLIFICATION OF THE SCALING PROCESS FOR MV PREDICTION

J. Dong, Y. Ye
InterDigital Communications LLC
July 2012

Motion Vector Scaling

- In HM7.0, MV prediction is performed based on temporal co-located PU and spatially neighbouring PUs



- Scaling based on POC distance is applied on the neighbouring MV

  $$MV2 = MV1 \quad (TB / TD)$$

- The following steps are applied using fixed point operations

1. Clip TB and TD to [-128, 127]
2. Calculate $iX=(2^{14}+ (Abs(TD)>>1))/TD$
3. Calculate $iScale = (iX \times TB+32)>>6$
4. Clip iScale to [-4096, 4095]
5. Calculate MV2

   $$MV2(x)=Sign(iScale \times MV1(x)) \times ((|iScale \times MV1(x)| + 127) >> 8)$$

   $$MV2(y)=Sign(iScale \times MV1(y)) \times ((|iScale \times MV1(y)| + 127) >> 8)$$

6. Clip MV2 to the range of -32768 to 32767, inclusive.

INTERDIGITAL.

Simplified iScale calculation

- In the current scheme, iScale is calculated in 4 steps
  - First, TB and TD are clipped to [-128, 127]
  - Second, iX ≈ 1 / TD in 14-bit representation

    $iX=(2^{14}+ (Abs(TD)>>1))/TD$
  - Third, iScale ≈ TB / TD in 8-bit representation

    $iScale = (iX \times TB+32)>>6$
- Proposal simplification:
  - Clip TB and TD to [-32, 31]
  - Calculate iScale from TB and TD

    $iScale = ((TB<<9) + Sign(TB) \times Abs(TD))/(TD<<1)$

INTERDIGITAL.

Comparison between HM7.0 and proposed

	HM7.0		Proposed	
	Operation	HW imp	Operation	HW imp
1	Clip TB and TD to [-128, 127]		Clip TB and TD to [-32, 31]	
2	iX=(2^{14}+ (Abs(TD)>>1))/TD	128×16-b LUT	–	–
3	iScale = (iX×TB+32)>>6	16-b × 8-b multiplier	iScale = ((TB<<9) + Sign(TB) × Abs(TD))/(TD<<1)	32×32×16-b LUT
4	Clip iScale to [-4096, 4095]		Clip iScale to [-4096, 4095]	
5	MV2=Sign(iScale × MV1) × ((|iScale × MV1| + 127) >> 8)	16-b × 13-b multiplier	MV2=Sign(iScale × MV1) × ((|iScale × MV1| + 127) >> 8)	16-b × 13-b multiplier
6	Clip MV2 to [-32768, 32767]		Clip MV2 to [-32768, 32767]	

- Benefits:
  - iScale calculated using 1 merged step instead of using 2 separate steps
  - Remove one 16-bit   8-bit multiplier
- Hardware implementation can be done using 32 32 16-bit LUT
  - Based on symmetry, LUT can be indexed by Abs(TB) and Abs(TD)
  - iScale = Sign(TB TD)   iScale', where iScale' is LUT entry indexed by Abs(TB) and Abs(TD)

Simulation results

	RA Main			RA HE10		
	Y	U	V	Y	U	V
Class A	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E						
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Simulation results

	RA Main			RA HE10		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

INTERDIGITAL.

Conclusions

- Simplification of MV scaling process is proposed in this contribution
  - Clip TB and TD to [-32, 31]
  - iScale calculation using 1 combined step
- Can be implemented in HW by 32x32x16-bit LUT
- Bit-exact results under common test conditions

INTERDIGITAL.