

Bi-prediction by single motion vector using decoder-side inter-reference ME

JCTVC-E216

Motoharu Ueda
JVC KENWOOD Holdings, Inc.

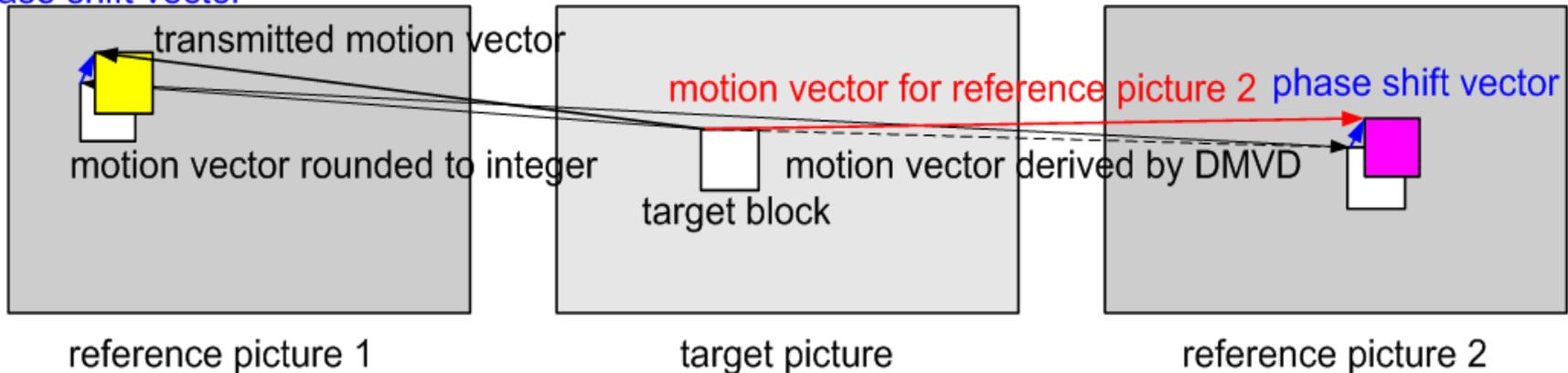
Overview

- Complexity analysis of DMVD
 - Focused on the technique “refinement motion compensation using DMVD”
 - Complexity measurements addition to the runtime measurement
- Simplified technique
 - DMVD search range
 - Interpolation filter for DMVD
 - Restrict to adapt PU size
- Experiments
 - JCTVC-D120 and the simplified algorithm are implemented to HM2.0
 - Performance and complexity comparison under random access over HM2.0
 - 1.5% BD-rate gains with 41% encode / 21% decode runtime increase
 - 1.2% BD-rate gains with 23% encode / 14% decode runtime increase
- Conclusions

Refinement motion compensation using DMVD

RMC was proposed and tested at previous meetings (JCTVC-A108, JCTVC-B032, JCTVC-C138, JCTVC-D120)

phase shift vector



MC block for reference picture 1

motion vector for reference picture 1 (MV1) : transmitted vector

(motion vector rounded to integer + phase shift vector)



MC block for reference picture 2

motion vector for reference picture 2 (MV2) :

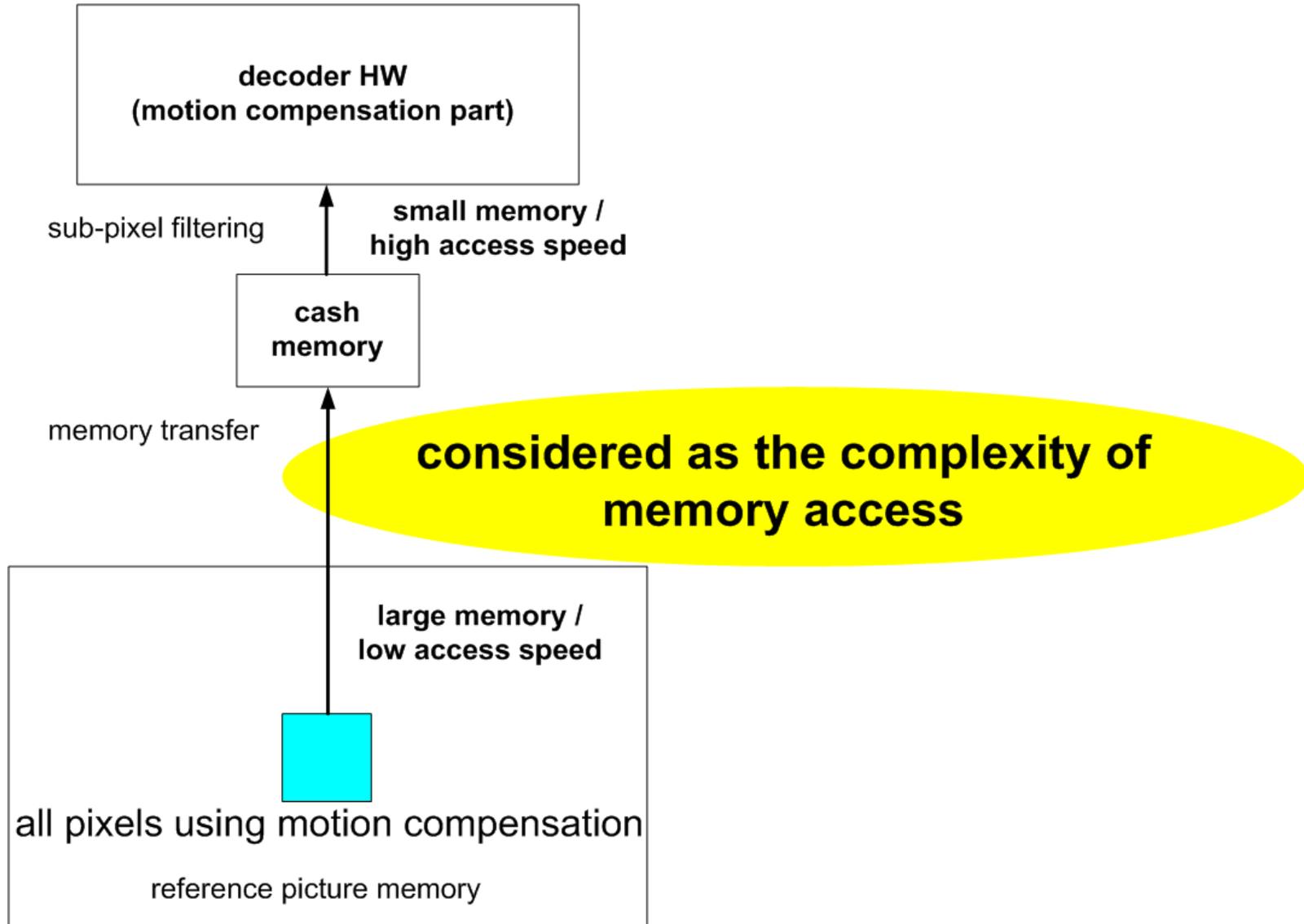
transmitted vector + motion vector derived by DMVD

(motion vector rounded to integer + motion vector derived by DMVD + phase shift vector)

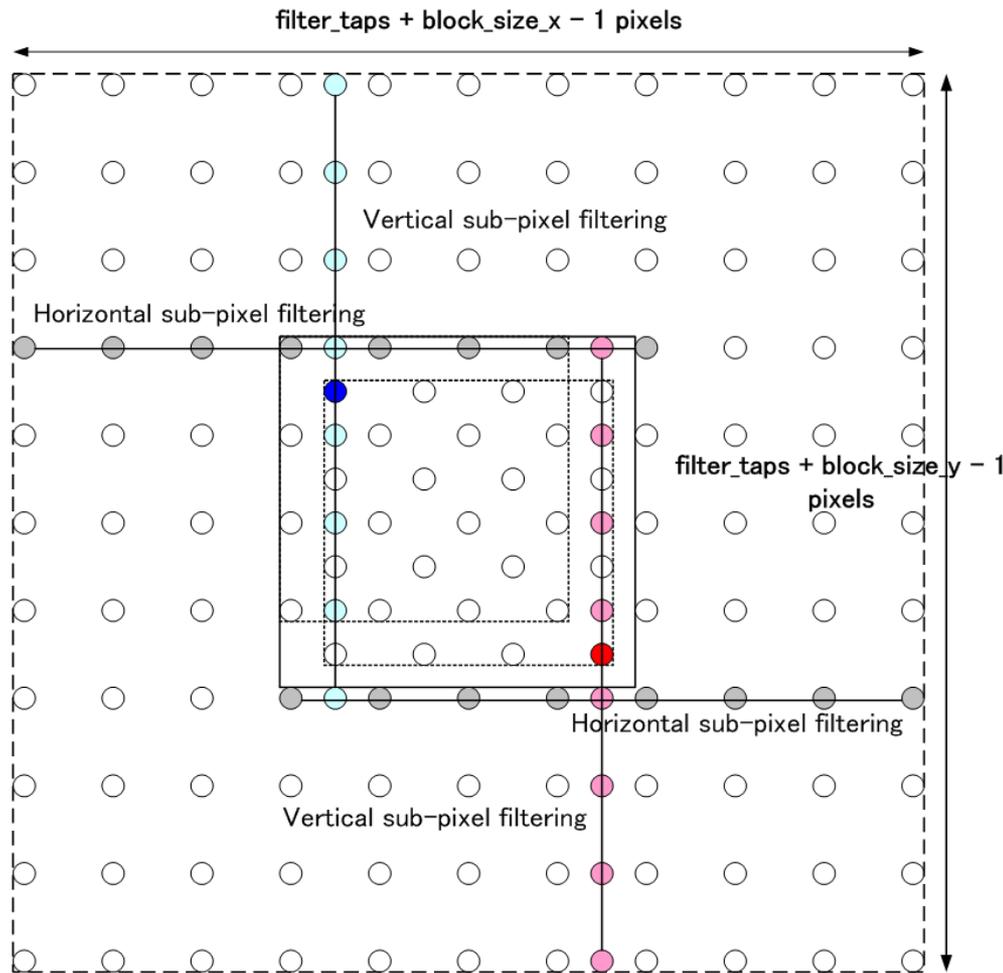
The additional complexity measurements

- Estimate the increase of the decoder complexity more exactly (in addition to runtime measurement)
 - The reference memory access for motion compensation
 - considering the worst case
 - The number of operations for DMVD
 - interpolation filter operation for sub-pixel search
 - block matching operation

The assumption model of motion compensation



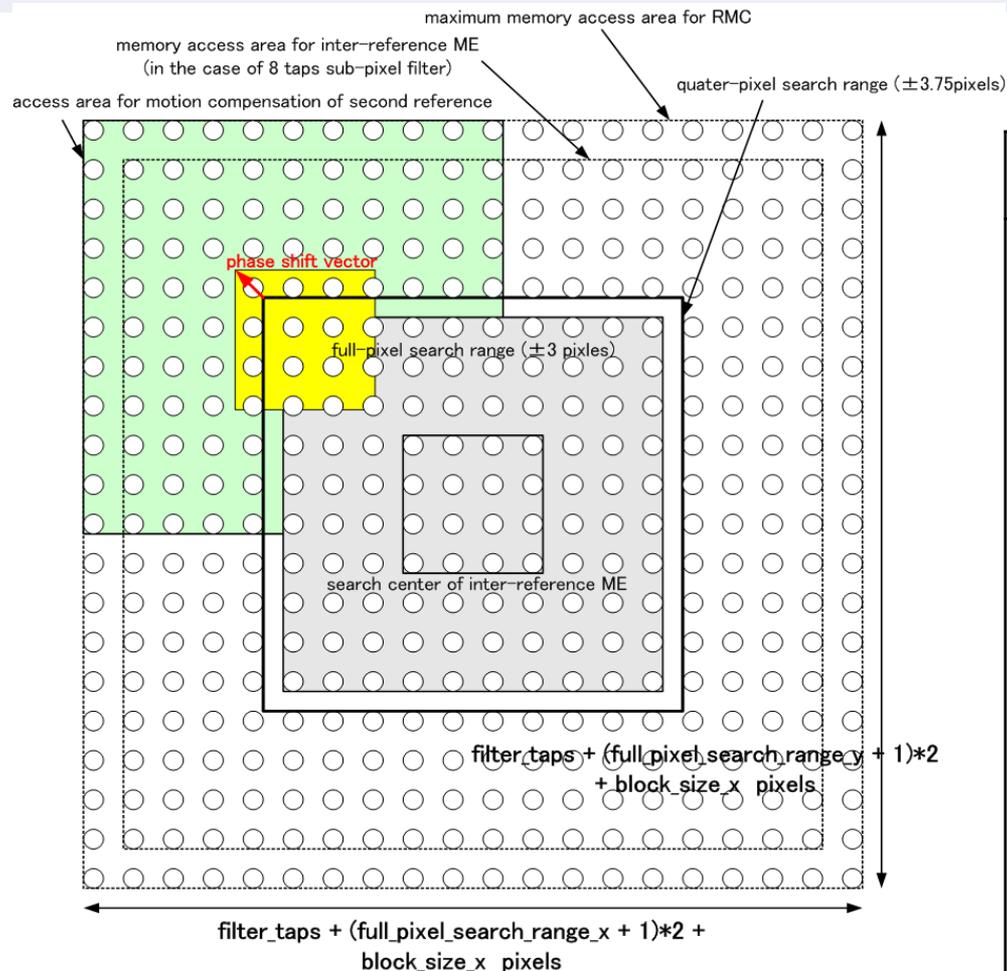
Reference memory access for MC



MC blocksize	max access ratio
4x4	15.13
4x8/8x4	10.31
8x8	7.03
8x16/16x8	5.39
16x16	4.13
16x32/32x16	3.5
32x32	2.97
32x64/64x32	2.7
64x64	2.46

- The necessary reference memory access area (NEMA)
 - $NEMA = (filter_taps + block_size_x - 1) * (filter_taps + block_size_y - 1)$

Reference memory access for RMC



MC blocksize	max access ratio
4x4	32.56
4x8/8x4	20.16
8x8	12.52
8x16/16x8	8.7
16x16	6.07
16x32/32x16	4.75
32x32	3.74
32x64/64x32	3.23
64x64	2.79

■ The necessary reference memory access area (NEMA_RMC)

■
$$\text{NEMA_RMC} = (\text{filter_taps} + (\text{full_pixel_search_range_x} + 1) * 2 + \text{block_size_x}) * (\text{filter_taps} + (\text{full_pixel_search_range_y} + 1) * 2 + \text{block_size_y})$$

The number of operations for DMVD

- DCTIF 8tap filter specification

position	coefficients (6bits) / 64	shift operation	add operation
1/4	{ -1, 4, -10, 57, 19, -7, 3, -1 }	4	12
1/2	{ -1, 4, -11, 40, 40, -11, 4, -1 }	4	9

- The number of operations for sub-pixel filtering of DMVD

- Half-pixel 96 shift , 216 add operations / pixel

- Quarter-pixel 96 shift , 288 add operations / pixel

- Total 696 operations / pixel

- The number of operations for block matching of DMVD

- Search point $49 + 8 + 8 = 65$ (JCTVC-D120 algorithm)

- SAD function 2 add, 1 absolute operations / pixel

- Total 190.9375 operations / pixel

The increase of complexity of DMVD

specification	current HEVC (HM2.0)	JCTVC-D120 ported to HM2.0
worst case of memory access ratio	15.13 (bi-pred in 4x4PU)	32.56 (RMC mode in 4x4PU)
number of operations for DMVD	0	886.9375 times / pixel

Simplified approach

- Approach toward decrease complexity of the decoder
 - In the aspect of memory access and number of operations
- **Limited search range of DMVD**
 - Decrease of the worst case of reference memory access
 - Ratio : from 32.56x to 23.56x
 - Decrease of the number of block matching operations
 - Operations : from 190.9375 to 73.4375 (about 100 times/pixel)
- **Bi-linear filter for DMVD sub-pixel filtering**
 - Decrease of the number of sub-pixel filtering operations
 - Operations : from 696 times to 60 times / pixel (90% decrease)
- **Restriction of the PU size**
 - Decrease of the worst case of reference memory access
 - Ratio : from 32.56x to 12.52x (smaller than normal MC for 4x4PU)

The specification of JCTVC-D120 and Simplified

Specification	Original RMC algorithm (same as JCTVC-D120)	Simplified RMC algorithm (decrease the complexity)
Search range of inter-reference ME	± 3.75 pixels	± 1.75 pixels
Sub-pixel ME accuracy	1/4 pixel	1/4 pixel
Interpolation filter for inter-reference ME	Same as MC (DCTIF 8tap)	bi-linear filter (2 tap)
Matching technique	SAD (luminance only)	SAD (luminance only)
DMVD block size	Same size as PU	Same size as PU
DMVD size limitation	Allowed all size of PU	Not used for 4x4/4x8/8x4PU
Worst case of memory access ratio	32.56 (RMC mode in 4x4PU)	15.13 (normal bi-pred in 4x4PU)
Number of operations for DMVD	886.9375 times / pixel	133.4375 times / pixel

- Implemented into HM2.0
 - The parameters send by the slice header

Simulation conditions

- Conditions
 - Common test conditions HM2.0
 - Random Access High Efficiency only

- Simulation environments

- Encoder

CPU	Intel Xeon 5160 3.00GHz
Memory	16GB / 12GB (DDR3)
OS	Windows 7 Professional 64bit
Compiler	Microsoft Visual C++ 2008 Professional edition SP1
Executable	x64

- Decoder

CPU	Intel Core i7 860 2.80GHz
Memory	8GB (DDR3)
OS	Windows 7 Professional 64bit
Compiler	Microsoft Visual C++ 2008 Professional edition SP1
Executable	X64

Performance comparison results (RA)

■ BD-rate (results of each test sequences) at Random Access

sequence	original (JCTVC-D120) algorithm			simplified algorithm		
	BD-rate Y	BD-rate U	BD-rate V	BD-rate Y	BD-rate U	BD-rate V
Traffic	-1.24	-0.91	-0.95	-1.07	-0.83	-0.82
PeopoleOnStreet	-2.25	-3.34	-3.03	-1.78	-2.32	-2.14
Nebuta	-0.44	-1.14	-1.09	-0.39	-1.41	-1.35
SteamLocomotive	-1.32	-1.62	-1.26	-1.11	-1.33	-1.46
Kimono	-1.42	-1.03	-1.08	-1.06	-0.82	-0.84
ParkScene	-0.88	-0.80	-0.85	-0.66	-0.52	-0.60
Cactus	-1.54	-1.51	-1.47	-1.47	-1.14	-1.22
BasketballDrive	-2.38	-2.15	-2.38	-1.77	-1.31	-1.64
BQTerrace	-1.14	-0.90	-0.79	-1.30	-0.50	-0.38
BasketballDrill	-1.93	-2.25	-2.25	-1.39	-1.43	-1.40
BQMall	-1.69	-1.30	-1.29	-1.32	-0.75	-0.90
PartyScene	-1.24	-1.50	-1.46	-0.98	-1.04	-0.94
RaceHorses	-2.19	-2.32	-2.56	-1.25	-1.21	-1.52
BasketballPass	-2.47	-2.65	-2.52	-1.69	-1.49	-1.40
BQSquare	-0.23	-0.43	-0.31	-0.33	-0.32	-0.37
BlowingBubbles	-1.13	-0.99	-1.01	-0.77	-0.48	-0.63
RaceHorses	-2.39	-2.75	-2.74	-1.39	-1.65	-1.67
ClassA Average	-1.3	-1.8	-1.6	-1.1	-1.5	-1.4
ClassB Average	-1.5	-1.3	-1.3	-1.3	-0.9	-0.9
ClassC Average	-1.8	-1.8	-1.9	-1.2	-1.1	-1.2
ClassD Average	-1.6	-1.7	-1.7	-1.1	-1.0	-1.0
Total Average	-1.5	-1.6	-1.6	-1.2	-1.1	-1.1

Average 1.5% gains for JCTVC-D120 and 1.2% gains for simplified algorithm

Complexity comparison results

- Runtime measurement
- Encoder : result of execution time (max 4 process on 4core CPU)
- Decoder : result of execution time (1 process)
 - with output of yuv file

	JCTVC-D120 algorithm			simplified algorithm		
	Y BD-rate	U BD-rate	V BD-rate	Y BD-rate	U BD-rate	V BD-rate
Enc Time[%]	142%			123%		
Dec Time[%]	121%			114%		

- 42% encode / 21% decode runtime at JCTVC-D120 algorithm
- 23% encode / 14% decode runtime at simplified algorithm

Additional results (independently adapted)

	DMVD search range = ± 1			bilinear filter for DMVD			No DMVD for 4x4/4x8/8x4PU		
	Y BD-rate	U BD-rate	V BD-rate	Y BD-rate	U BD-rate	V BD-rate	Y BD-rate	U BD-rate	V BD-rate
Class A	-1.3	-1.8	-1.8	-1.3	-1.9	-1.7	-1.3	-1.8	-1.9
Class B	-1.4	-1.1	-1.1	-1.4	-1.2	-1.2	-1.4	-1.2	-1.2
Class C	-1.6	-1.6	-1.6	-1.6	-1.7	-1.8	-1.6	-1.6	-1.7
Class D	-1.5	-1.4	-1.5	-1.4	-1.5	-1.5	-1.3	-1.3	-1.3
Class E									
All	-1.4	-1.4	-1.5	-1.4	-1.5	-1.5	-1.4	-1.5	-1.5
Enc Time[%]		140%			139%			125%	
Dec Time[%]		117%			117%			122%	
worst case of memory access ratio		23.56 (RMC mode in 4x4PU)			32.56 (RMC mode in 4x4PU)			15.13 (normal bi-pred in 4x4PU)	
number of operations for DMVD		769.4375 times / pixel			250.9375 times / pixel			886.9375 times / pixel	

Conclusion

- Proposed technique
 - Simplified RMC algorithm considering the additional complexity measurements - reference memory access, block matching operations
- Performance comparison under Random Access
 - 1.5% gains at JCTVC-D120, 1.2% gains at simplified algorithm
- Complexity comparison under Random Access
 - 42% encode / 21% decode runtime at JCTVC-D120 algorithm
 - 23% encode / 14% decode runtime at simplified algorithm
 - The additional complexity measurements of simplified algorithm
 - The worst case of memory access is the same as HM2.0
 - 85% decrease of operations for DMVD against JCTVC-D120
- Future Work
 - Investigate RMC with other DMVD-related tools in CE
 - Complexity with “runtime”, “additional measurements”, “parallelism”

JVC KENWOOD HOLDINGS

