

JCTVC-E206

In-loop filter based on non-local means filter

Masaaki Matsumura, Yukihiro Bandoh,
Seishi Takamura and Hirohisa Jozawa
NTT Cyber Space Laboratories
NTT Corporation, Japan



Summary

□ Background

- Current in-loop filter design of HM2.0
 - Deblocking filter (DF): similar to H.264/AVC
 - Adaptive loop filter (ALF): Wiener filter based
- These are kind of local filter

□ Proposal

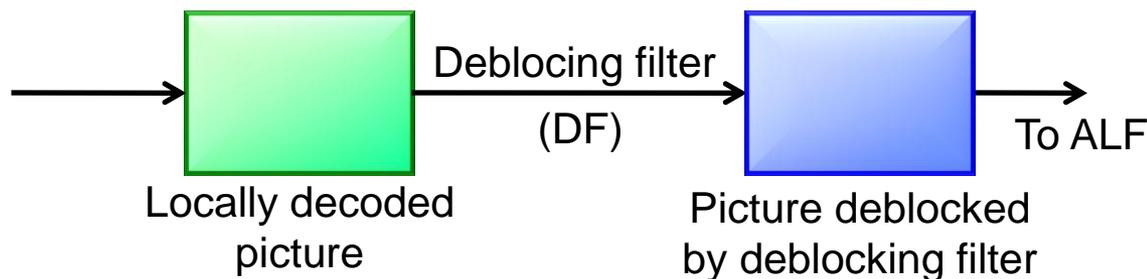
- Denoising filter based on non-local means (NLM) operation as a part of the in-loop filter

□ Results

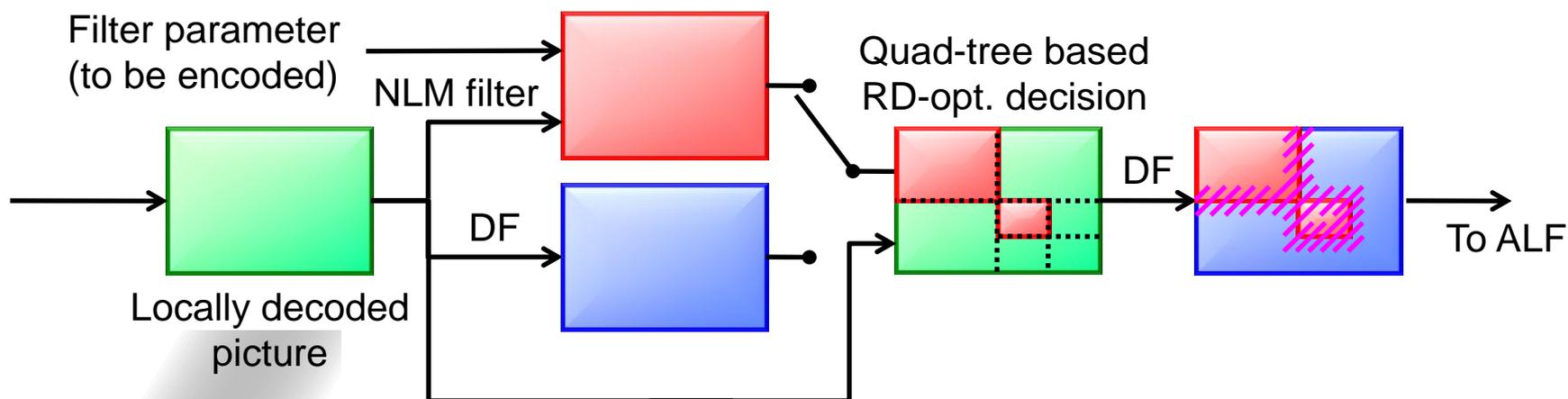
- 1.6 - 2.9% BD-rate (Y) improvements on average
- 4.1% BD-rate (Y) improvements at maximum

Proposed in-loop filter

Conventional in-loop filter



Proposed in-loop filter

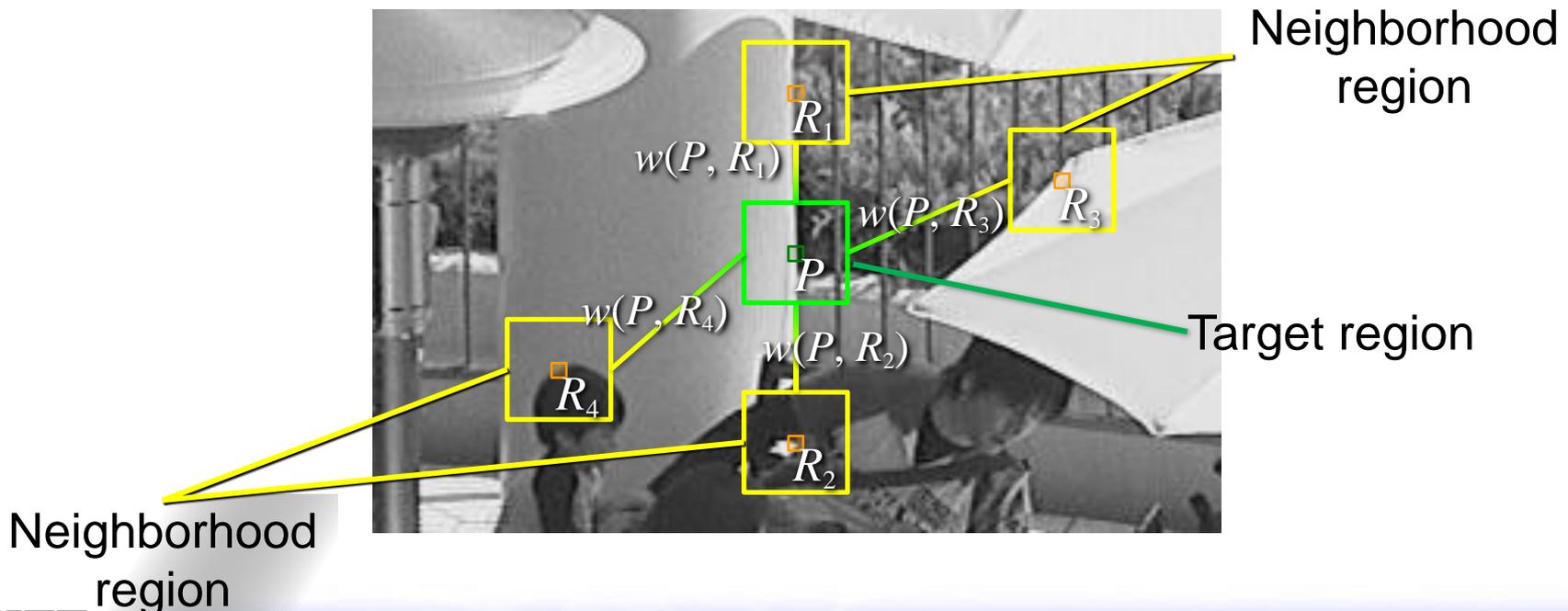


Non-local means (NLM) filter

Shift-variant filter based on a similarity between regions

Idea: Regions with high *similarity* are assigned large weights

Similarity: SSD between the target and neighborhood



Performance (All Intra)

Anchor: HM2.0-r598 (high efficiency).

| | Intra | | |
|-----------------------------|-----------|-----------|-----------|
| | Y BD-rate | U BD-rate | V BD-rate |
| Class A | -1.7 | -1.0 | -0.9 |
| Class B | -1.3 | -1.3 | -2.0 |
| Class C | -2.1 | -2.6 | -3.0 |
| Class D | -1.3 | -2.1 | -3.1 |
| Class E | -1.8 | 0.6 | 0.5 |
| All | -1.6 | -1.4 | -1.8 |
| Enc Time[%] | 145% | | |
| Dec Time[%] (single-thread) | 152% | | |
| Dec Time[%] (multi-thread) | 126% | | |

Performance (RA)

Anchor: HM2.0-r598 (high efficiency).

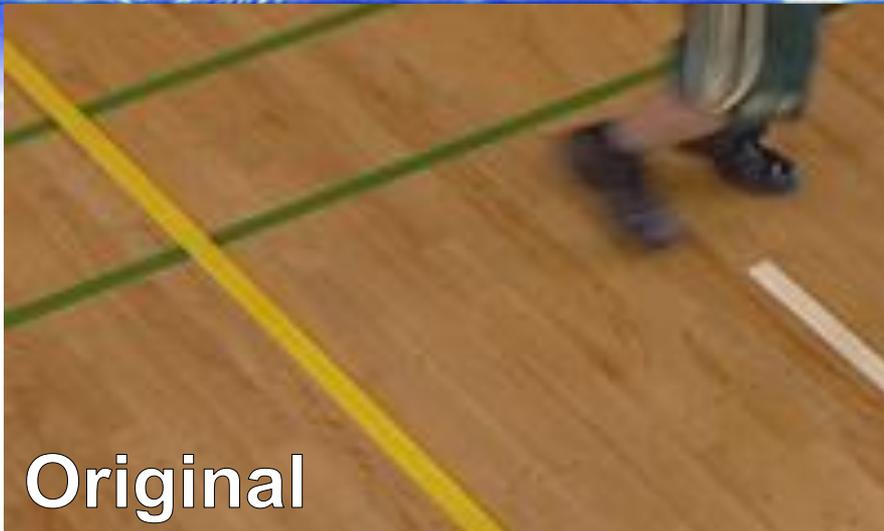
| | Random access | | |
|-----------------------------|---------------|-----------|-----------|
| | Y BD-rate | U BD-rate | V BD-rate |
| Class A | -1.6 | -2.0 | -1.9 |
| Class B | -2.1 | -2.6 | -2.1 |
| Class C | -2.3 | -4.1 | -4.0 |
| Class D | -1.5 | -2.5 | -3.2 |
| Class E | | | |
| All | -1.9 | -2.8 | -2.8 |
| Enc Time[%] | 118% | | |
| Dec Time[%] (single-thread) | 124% | | |
| Dec Time[%] (multi-thread) | 114% | | |

Performance (LD)

Anchor: HM2.0-r598 (high efficiency).

| | Low delay | | |
|-----------------------------|-----------|-----------|-----------|
| | Y BD-rate | U BD-rate | V BD-rate |
| Class A | | | |
| Class B | -2.3 | -2.5 | -2.9 |
| Class C | -3.1 | -4.8 | -5.3 |
| Class D | -2.7 | -2.1 | -2.8 |
| Class E | -4.1 | 0.1 | -1.8 |
| All | -2.9 | -2.5 | -3.3 |
| Enc Time[%] | 112% | | |
| Dec Time[%] (single-thread) | 131% | | |
| Dec Time[%] (multi-thread) | 117% | | |

Subjective quality



BasketballDrill, LD, QP=37,
BD-rate: -4.1%(Y), -8.8%(U),
-8.9%(V)
Frame #499



- ❑ Reduces mosquito noise
- ❑ Preserves chroma
(exp. lines on the floor)

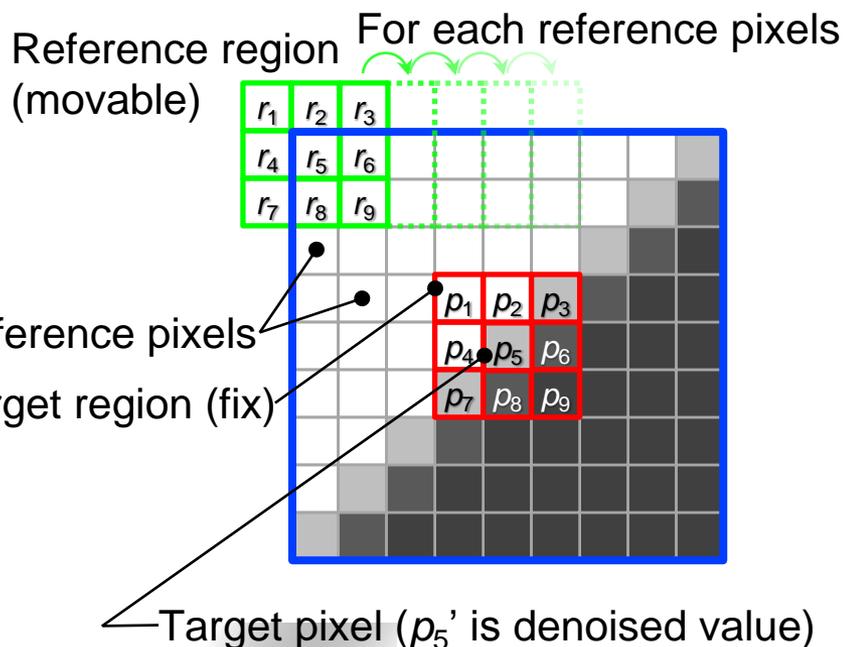
Conclusion

- A denoising filter as a part of the in-loop filter is presented
- Results
 - 1.6 - 2.9% BD-rate (Y) improvements on average
 - 4.1% BD-rate (Y) improvements at maximum
 - Enhanced subjective quality
 - EncTime: 112-145%, DecTime: 124-152% (Parallel processing friendly -- DecTime 114-126%)
 - Further speeding up has been pursued
- Proposal
 - To be studied in CE (current CE8 or CE12) to find the optimal combination with DF/ALF

Thank you, questions?

Basic algorithm of NLM

NLM filter: non-linear operation based on a similarity between pixel neighborhoods.
Basic idea: pixel neighborhoods with high similarity provide large weight for filtering.
Similarity: SSD between the neighborhood of target pixel and the neighborhood area in the reference region.



initialize: $SumWeights = 0, SumPixels = 0$

for all reference pixels do

$$Weight = \exp\left(\sum_{n=1}^9 (r_n - p_n)^2 / DenoiseParameter\right)$$

$$SumWeights = SumWeights + Weight$$

$$SumPixels = SumPixels + Weight \times r_5$$

end for

$$p_5' = SumPixels / SumWeights$$

Note: *DenoiseParameter* is negative value