Decoder-side Block Boundary Decision (DBBD) with OBMC

JCTVC-C137
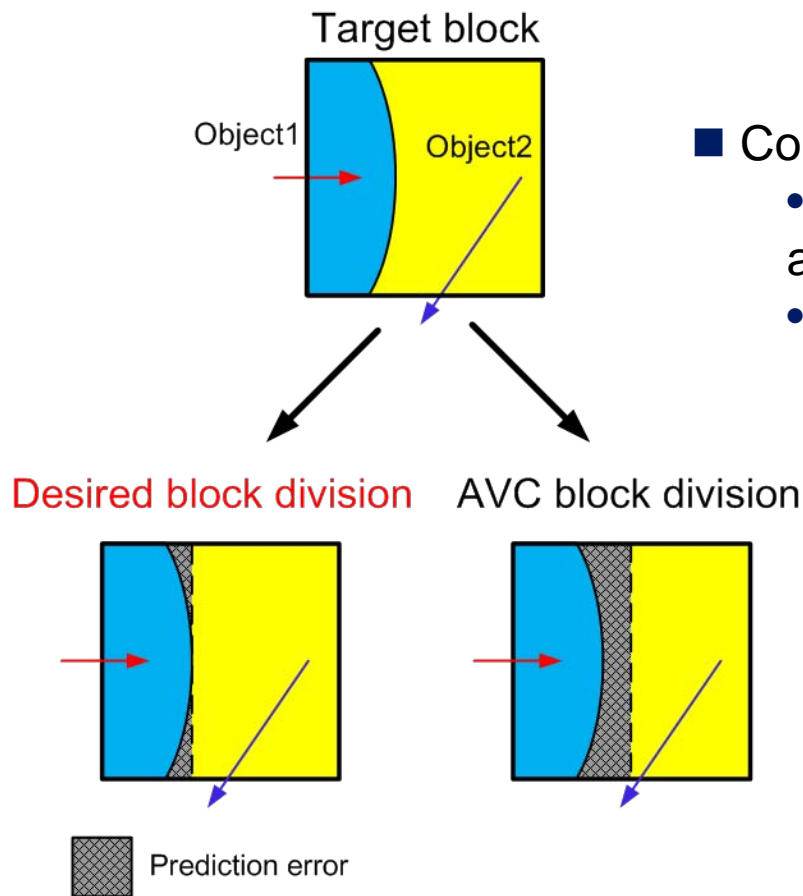
Shigeru Fukushima
JVC KENWOOD Holdings, Inc. (JVC)

JVC KENWOOD Holdings, Inc.
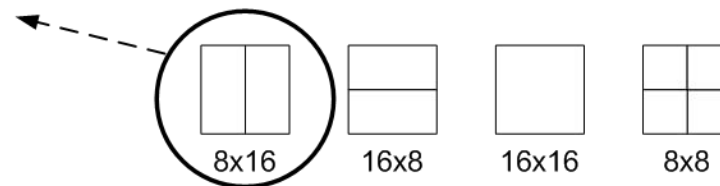
1. Outline

Outline

- **Proposed technique**

    - DBBD (Decoder-side Block Boundary Decision) with OBMC

    - DBBD was proposed in the JCTVC-A108

- **About DBBD**

    - Inter prediction technique

    - Exploit the correlation between actual motion boundary and object boundary

- **Implementation**

    - JM16.2 base

    - Same syntax as AVC perfectly

    - Implemented only to 16x8 and 8x16 mode

- **BD-rate**

    - Average gain 3.1% for Random Access

    - Average gain 2.2% for Low Delay

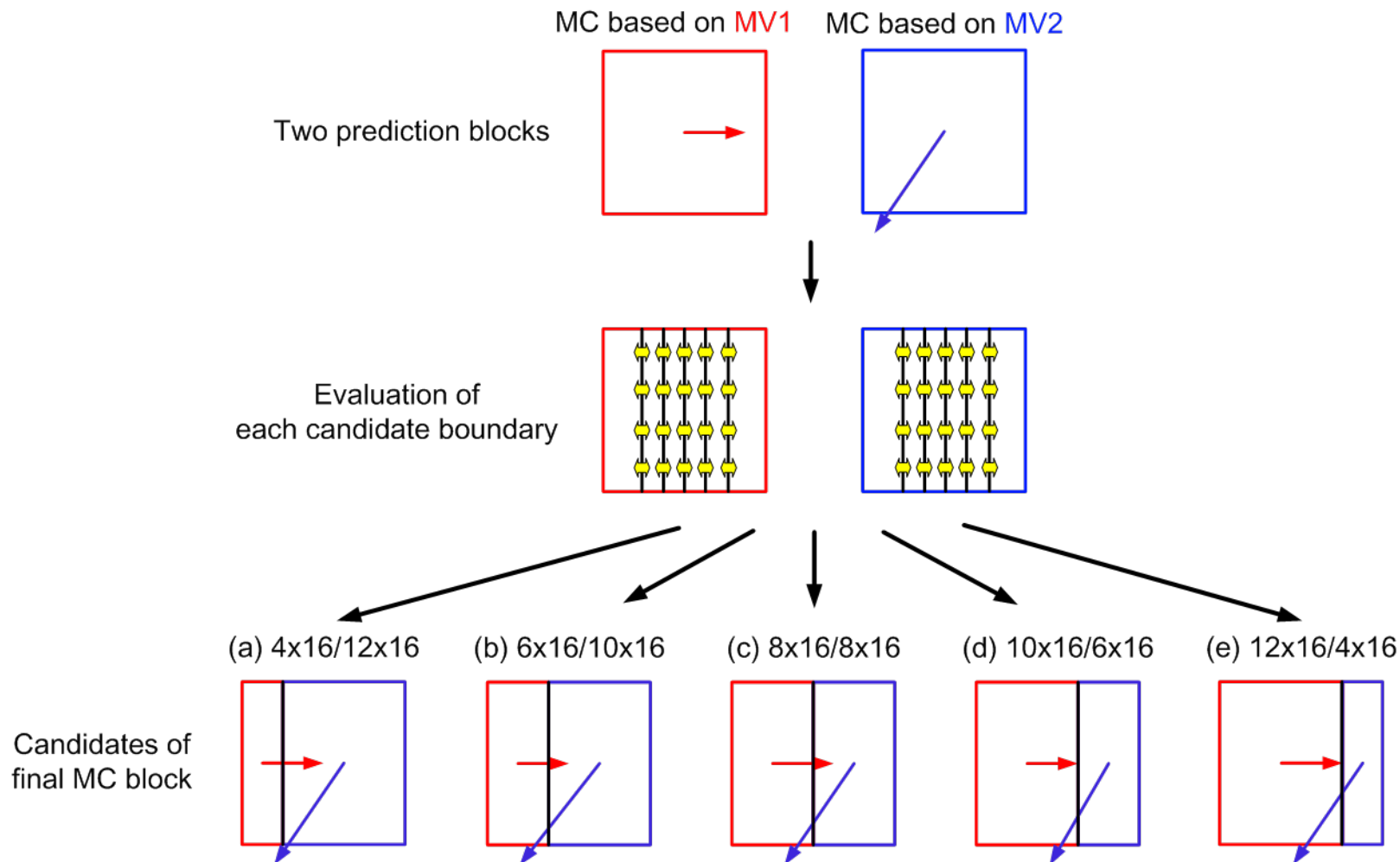# 2. DBBD Algorithm

Target block

Object1

Object2

■ Conventional techniques
- **mismatch** between the block boundary and actual motion boundary
- the **additional coding bits** required

Desired block division   AVC block division
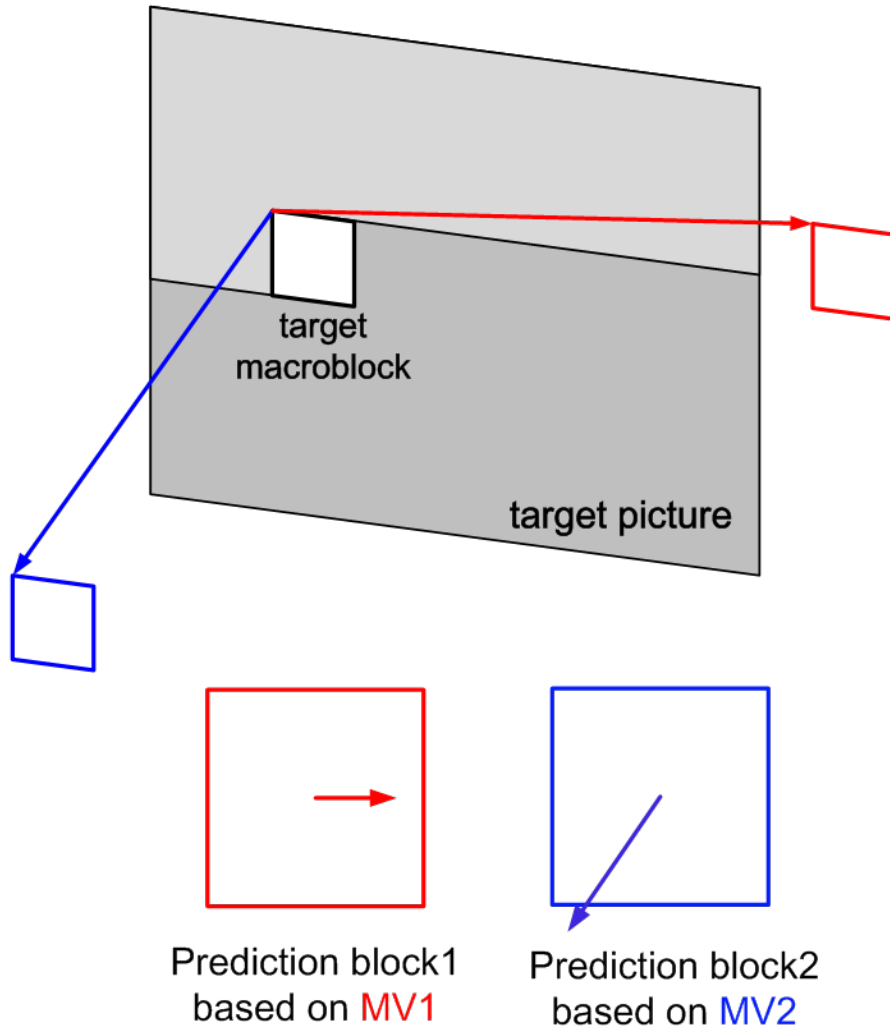
Prediction error

8x16   16x8   16x16   8x8
Fixed block patterns in AVC

It is desirable to support **various block patterns without additional coding bits** for MC.

MC with various block patterns without additional coding bits
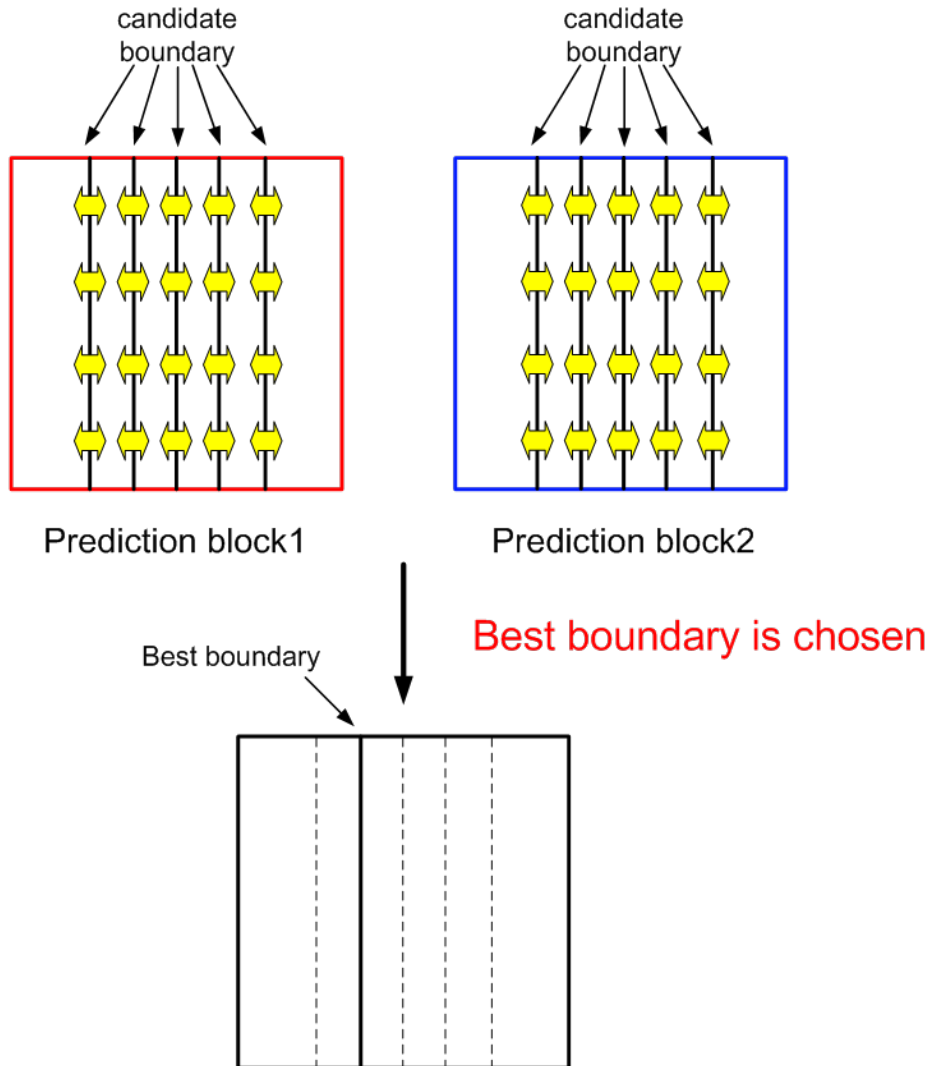- the block boundary is decided **on the decoder-side**

MC based on MV1   MC based on MV2

Two prediction blocks

Evaluation of
each candidate boundary

(a) 4x16/12x16   (b) 6x16/10x16   (c) 8x16/8x16   (d) 10x16/6x16   (e) 12x16/4x16

Candidates of
final MC block

target
macroblock

target picture

Prediction block1
based on MV1

Prediction block2
based on MV2

■ **Decoding procedure**

1. Generate two prediction blocks based on two MVs by the size of macroblock.

2. Evaluate each candidate boundary in order to decide the boundary.

3. Perform OBMC along the decided boundary using two prediction blocks.

4. Obtain the final MC block by replacing the pixels around the boundary with OBMC filtered samples.

candidate boundary

candidate boundary

Prediction block1

Prediction block2

Best boundary

Best boundary is chosen

■ **Decoding procedure**

1. Generate two prediction blocks based on two MVs by the size of macroblock.

2. Evaluate each candidate boundary in order to decide the boundary.

3. Perform OBMC along the decided boundary using two prediction blocks.

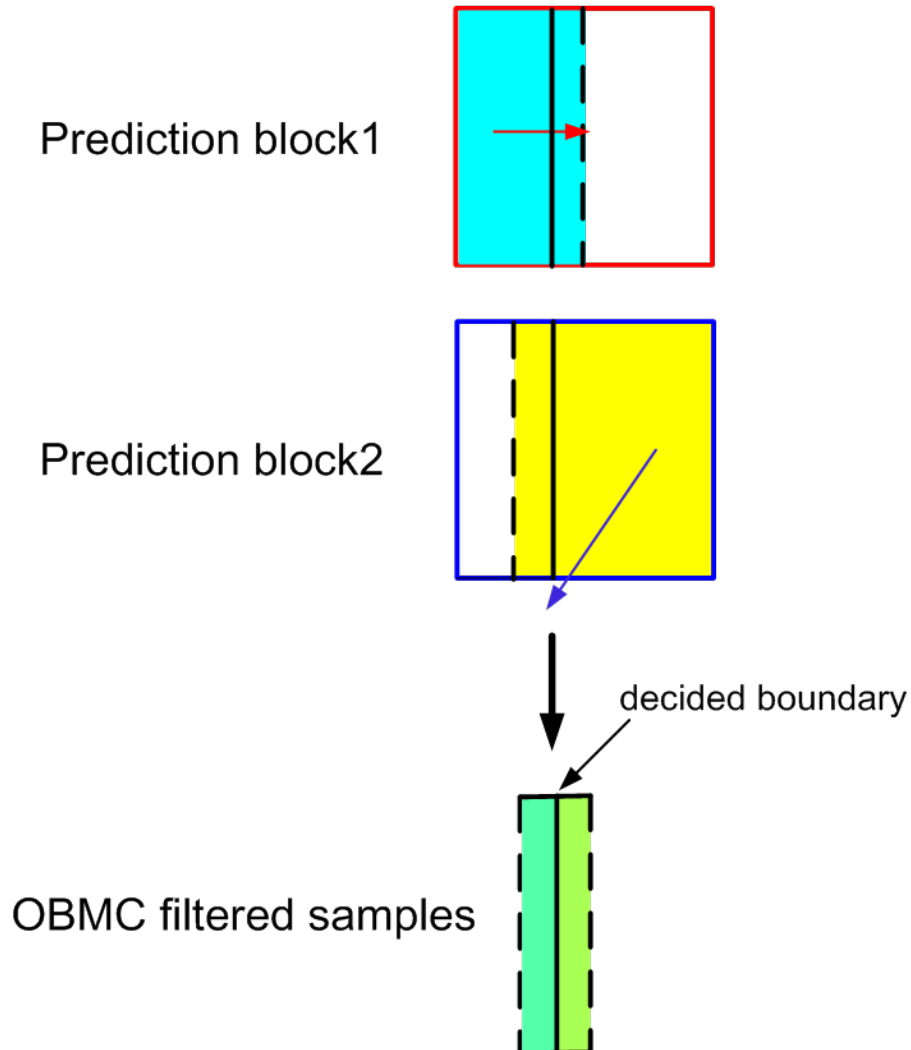4. Obtain the final MC block by replacing the pixels around the boundary with OBMC filtered samples.

Prediction block1

Prediction block2

decided boundary

OBMC filtered samples

■ **Decoding procedure**

1. Generate two prediction blocks based on two MVs by the size of macroblock.

2. Evaluate each candidate boundary in order to decide the boundary.

3. Perform OBMC along the decided boundary using two prediction blocks.

4. Obtain the final MC block by replacing the pixels around the boundary with OBMC filtered samples.
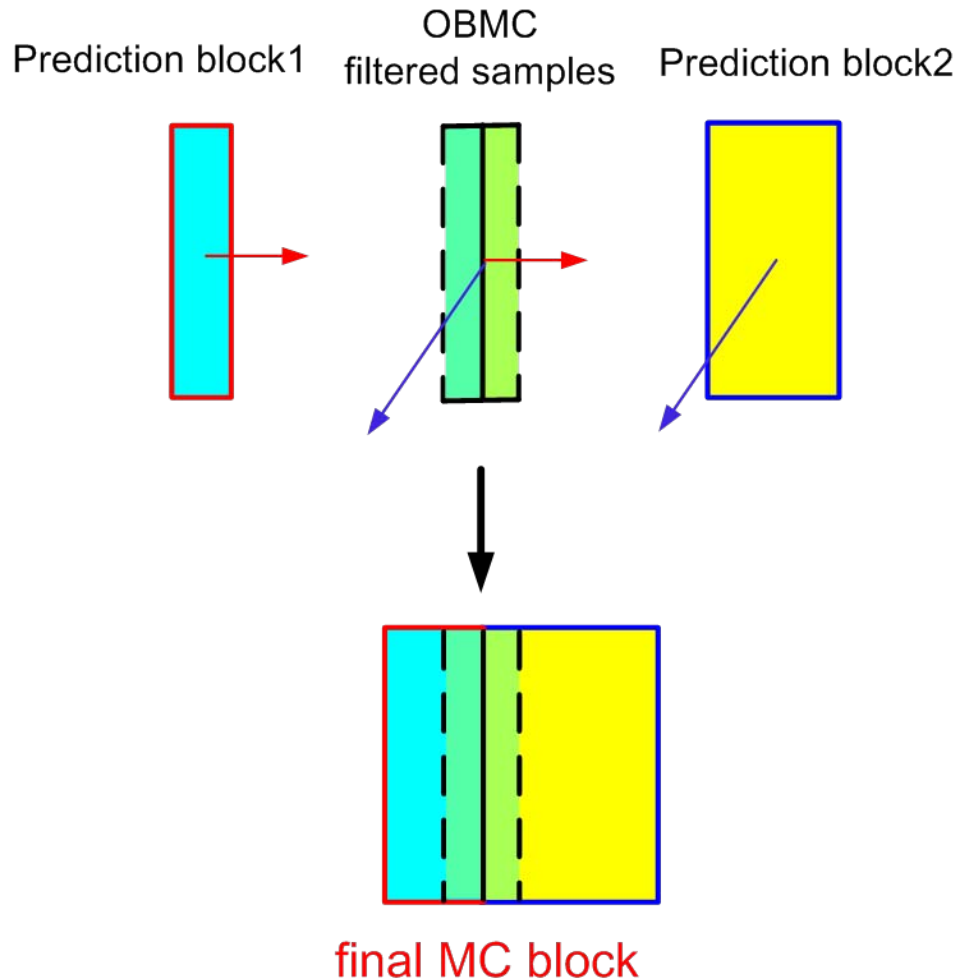
Prediction block1 — OBMC filtered samples — Prediction block2
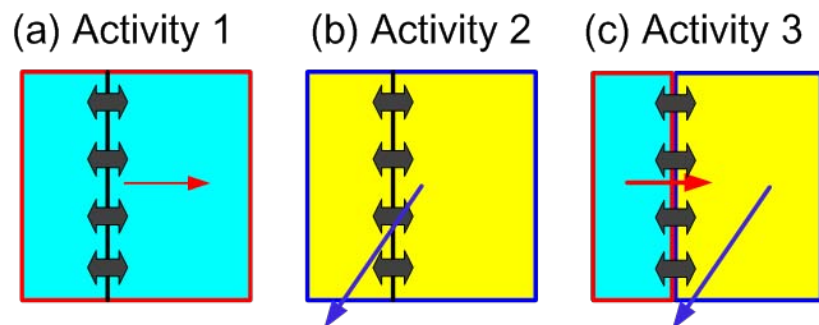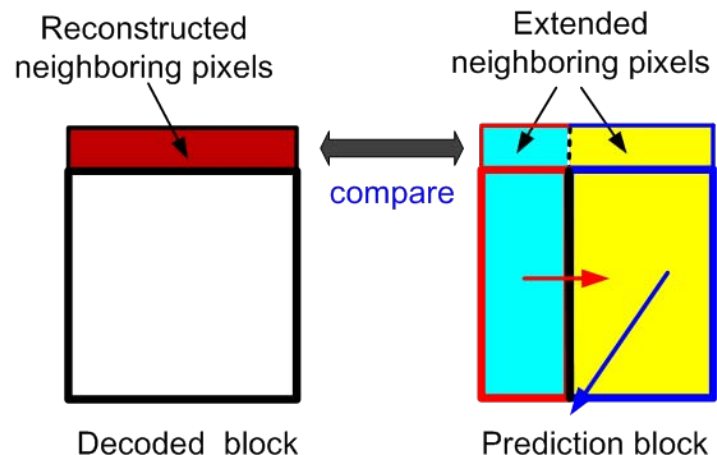
final MC block

■ **Decoding procedure**

1. Generate two prediction blocks based on two MVs by the size of macroblock.

2. Evaluate each candidate boundary in order to decide the boundary.

3. Perform OBMC along the decided boundary using two prediction blocks.

4. Obtain the final MC block by replacing the pixels around the boundary with OBMC filtered samples.

Boundary Evaluation

How to evaluate the boundary?
- calculate three Activities across the candidate boundary
- measure NEV (Neighboring Efficiency Value)

(a) Activity 1 (b) Activity 2 (c) Activity 3

(d) NEV (Neighboring Efficiency Value)

Reconstructed neighboring pixels

Extended neighboring pixels

compare

Decoded block

Prediction block

Boundary evaluation elements

(a) on prediction block 1 by MV1
(b) on prediction block 2 by MV2
(c) on connected two prediction blocks by MV1 and MV2
(d) compare extended prediction blocks with reconstructed pixels in neighboring region

Element	Activity1	Activity2	Activity3	NEV
Better value	Larger	Larger	Smaller	Smaller

OBMC

When?
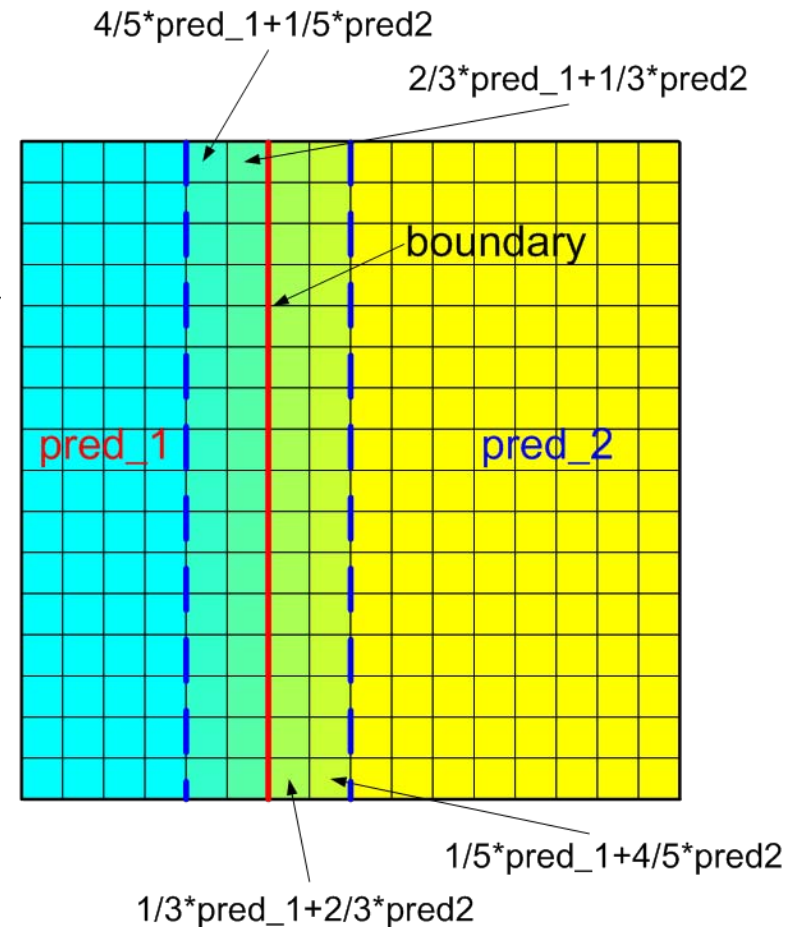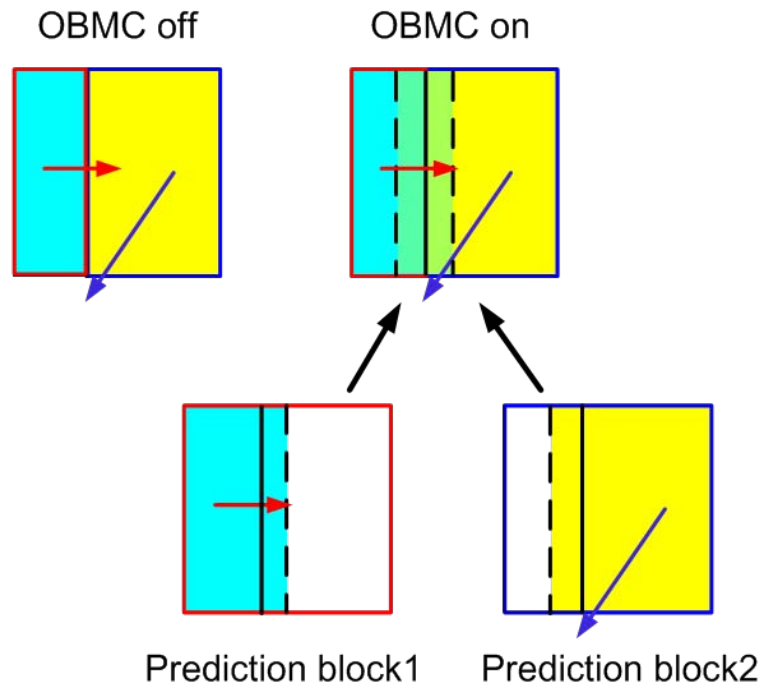- after decision of the boundary

Why?
- improve the quality of MC block

How?
- overlap two MC blocks along the boundary

OBMC off   OBMC on

Prediction block1   Prediction block2

4/5*pred_1+1/5*pred2

2/3*pred_1+1/3*pred2

boundary

pred_1   pred_2

1/5*pred_1+4/5*pred2

1/3*pred_1+2/3*pred2

Implementation

- Software

    - JM16.2

- Syntax

    - Same as AVC perfectly

- Prediction Mode

    - Applied only to 16x8 and 8x16 mode as 16xN and Nx16, respectively

- PMV

    - Same as AVC

- Transform Size

    - Same as AVC (only 8x8 or 4x4 used)

- Others

    - The candidate boundary is restricted to 4, 6, 8, 10, and 12

    - Only luma prediction samples are used to evaluate the boundary

3. Simulation Results

Test Conditions

Match with HEVC common test conditions as much as possible

- Test Sequences
  - All common test sequences are used
  - All frames are encoded
- Coding Structure
  - Random Access and Low Delay
  - I/P/B frame for Random Access
  - I/B frame for Low Delay
- Reference Frames
  - Two reference frames on P-frame
  - One reference frame for each list on B-frame
- Quantization parameter
  - 22, 27, 32, and 37 for I-frame

BD-rate Results

■ Average gain 3.1% and up to 4.9% for Random Access

■ Average gain 2.2% and up to 3.3% for Low Delay

Class	Test Sequence	Random Access	Low Delay
Class A	Traffic	-2.8	-
	PeopleOnStreet	-4.9	-
Class B	Kimono	-1.8	-1.0
	ParkScene	-2.3	-1.6
	Cactus	-2.6	-2.3
	BasketballDrive	-3.0	-2.3
	BQTerrace	-3.2	-2.0
Class C	BasketballDrill	-3.2	-2.8
	BQMall	-3.7	-3.0
	PartyScene	-2.2	-1.4
	RaceHorses	-4.3	-3.0
Class D	BasketballPass	-3.0	-2.3
	BQSquare	-2.6	-1.9
	BlowingBubbles	-2.2	-1.9
	RaceHorses	-4.3	-3.0
Class E	Vidyo1	-	-1.8
	Vidyo3	-	-3.3
	Vidyo4	-	-1.9
Average		-3.1	-2.2

4. Conclusion

Conclusion

- **Proposed technique**
  - DBBD (Decoder-side Block Boundary Decision) with OBMC

- **Software**
  - JM16.2 base

- **BD-rate**
  - Average gain 3.1% for Random Access
  - Average gain 2.2% for Low Delay

- **Future Work**
  - Confirm the effectiveness of DBBD on TMuC/TM software
  - Participate in TE/CE as a proponent