INTER-BLOCK DEPENDENCIES CONSIDERATION FOR INTRA CODING IN H.264/AVC AND HEVC STANDARDS

M.Bichon†,*, J.Le Tanou†, M.Ropert†, W.Hamidouche*, L.Morin*, L.Zhang*

†ERICSSON TV and Media

*INSA de Rennes (IETR)

I. Context & Objective

- **Recent MPEG Intra Coding tools**
  - Prediction use reconstructed samples
  - Context Adaptive Binary Arithmetic Coding (CABAC)

  By design, these causal processes introduce *Inter-Blocks Dependencies* (see **B.**)

Context

- **Rate-Distortion Optimization (RDO)**
  - Coding parameters : $\vec{p}$
  - Distortion D and Rate R
  - Minimize R-D cost function:

    $minimize\ D(\vec{p}), s.t.\ R(\vec{p}) \leq R_T$
  - Turned into Lagrangian cost: $J(\vec{p})$

    $\min_{\vec{p}} J(\vec{p}) = \min_{\vec{p}} \left(D(\vec{p}) + \lambda . R(\vec{p})\right)$

Objective

- **Classical-RDO in Video Compression**
  - Partition a frame F into blocks of pixels
  - Common assumption:

  Each block i is <u>independent</u> from others

$$\min_{\vec{p}} J(\vec{p}) \approx \sum_{i \in F} \min_{\vec{p_i}} J_i(\vec{p_i})$$

- **Proposed model: Joint RDO**
  - For each block i, within the frame F
  - Consider Inter-Block Dependencies:

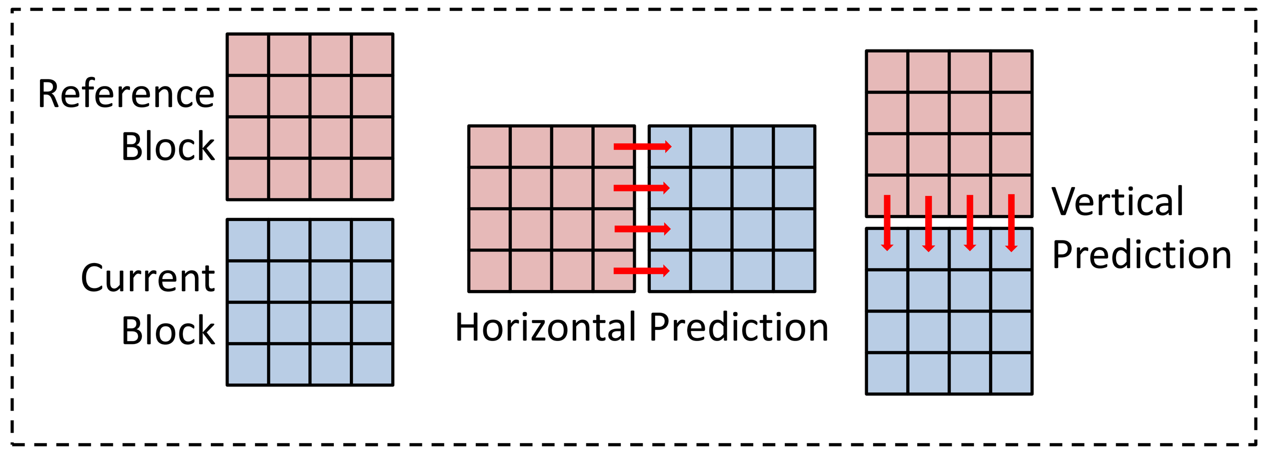
  Each block i is <u>dependent</u> from close ones

$$\min_{\vec{p}} J(\vec{p}) \approx \sum_{i \in F} \min_{\vec{p_i}} \left(J_i(\vec{p_i}) + J_{i+1}(\vec{p_i}, \vec{p_{i+1}})\right)$$

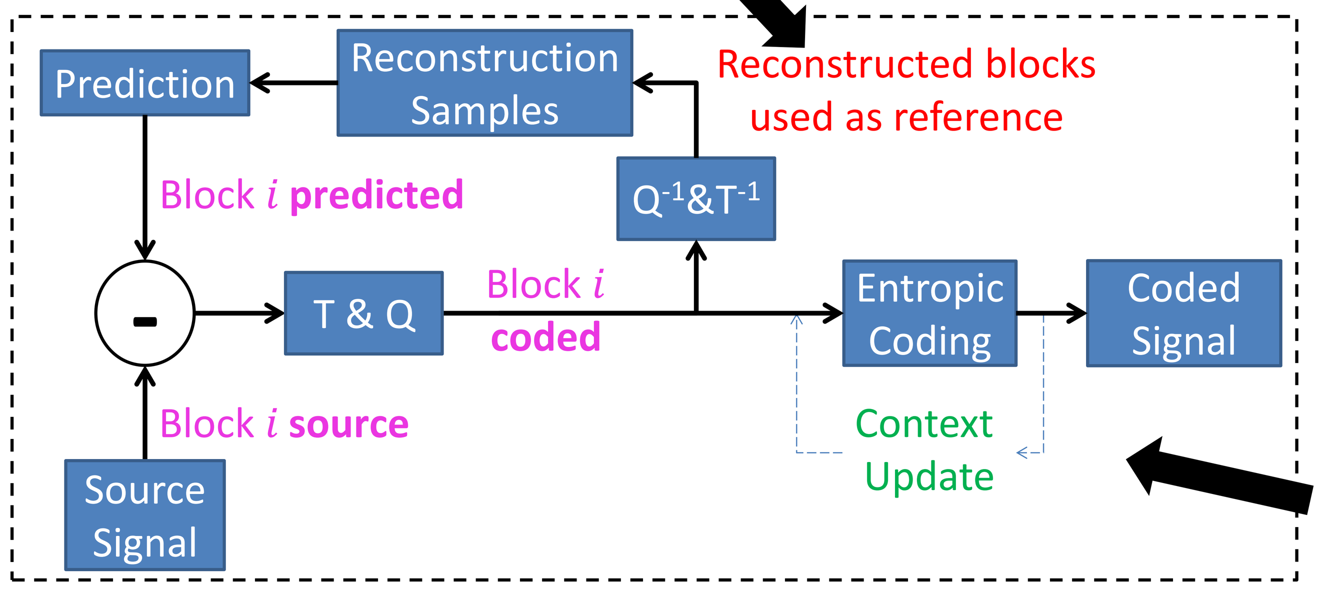
II. Inter-Block Dependencies

1. Intra Prediction in MPEG standards

- Spatial <u>projection</u> of neighboring blocks (+DC/Planar)
- Reference pixels are <u>reconstructed</u>



Distortion of one block may affect prediction of next ones



2. CABAC and Syntax Elements (*SE*)

- *SE* are binarized into symbols of n *bins*
- A context modeling provides estimates of conditional probabilities of each symbol bins
- *bin* coding is dependent of probability model updates from previous coded *bins*

Syntax bits depends on previously coded syntax elements

III. Proposed Joint RDO Models

- **Design of JRDO models is constrained**
  - Causal relationship between blocks
    - Raster Scan order and Z-Scan order
  - Computational complexity may be intractable
    - If $\vec{p}$ can take K different values
    - If N blocks are optimized by group of M blocks
    - Complexity turns from $(N.K)$ to $\left(\frac{N}{M}.K^M\right)$

Increasing M is assumed to be more efficient in terms of R-D cost, but also exponentially more complex

- **Application case**
  - Consider two cases:
    - $M = 2$ (**Dual-JRDO**) and $M = 4$ (**Quad-JRDO**)
  - Optimize prediction mode parameter (i.e. $\vec{p}$)
    - HEVC: $K = 35$
    - H.264/AVC: $K \in \{4,9\}$

- **Dual-JRDO**
  - Optimize blocks 2 by 2 (dotted areas)
  - High dependencies due to spatial proximity and coding orders

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

$$\{p_i^*, p_{i+1}'\} = arg \min_{\{p_i, p_{i+1}\}} \left(J_i(p_i) + J_{i+1}(p_i, p_{i+1})\right)$$

- **Quad-JRDO**
  - Optimize blocks 4 by 4
  - Better consideration of distortion propagation among all prediction modes (e.g. Vertical)

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

$$\{p_k^*\}_{k=i}^{i+3} = arg \min_{\{p_k\}_{k=i}^{i+3}} \sum_{k=i}^{i+3} J_i\left(\{p_l\}_{l=i}^{k}\right)$$

IV. Performance

- **Test Configuration**
  - Anchors in RDO:
    - HEVC (HM16.6)
    - H.264/AVC (JM19.0)
  - Configuration is All-Intra
  - First frame of each sequence
  - In HM, Quad-JRDO only apply to 4x4 blocks

Average BD-BR gains	JM19.0		HM16.6	
	Dual-JRDO	Quad-JRDO	Dual-JRDO	Quad-JRDO
Class B	-0.80%	-1.84%	-0.49%	-0.79%
Class C	-0.89%	-1.89%	-0.90%	-1.90%
Class D	-0.50%	-1.51%	-0.93%	-1.98%
Class E	-0.89%	-1.89%	-0.52%	-1.33%
All	**-0.77%**	**-1.78%**	**-0.71%**	**-1.47%**
Best Sequence	**-1.37%**	**-3.10%**	**-1.31%**	**-2.31%**
Worst Sequence	0.08%	-1.09%	-0.21%	-0.04%

- **Observations**
  - The more blocks are optimized concurrently, the more BD-BR reduction is achieved
  - Achievable gains of dependencies consideration are exhibited
  - **Jointly optimizing prediction modes brings systematic and substantial bitrate savings**

- **Future Work**
  - Different coding parameters can be optimized: QP, partitioning, lambda, …
  - Extension to this work to temporal dependencies is envisaged