Analysis of Multi-core Processing approaches

Regular Slices, Entropy Slices, Interleaved Entropy Slices, Wavefront Parallel Processing, Tiles

(JCTVC-F135/m20533)

Vivienne Sze, Madhukar Budagavi, Minhua Zhou

Texas Instruments

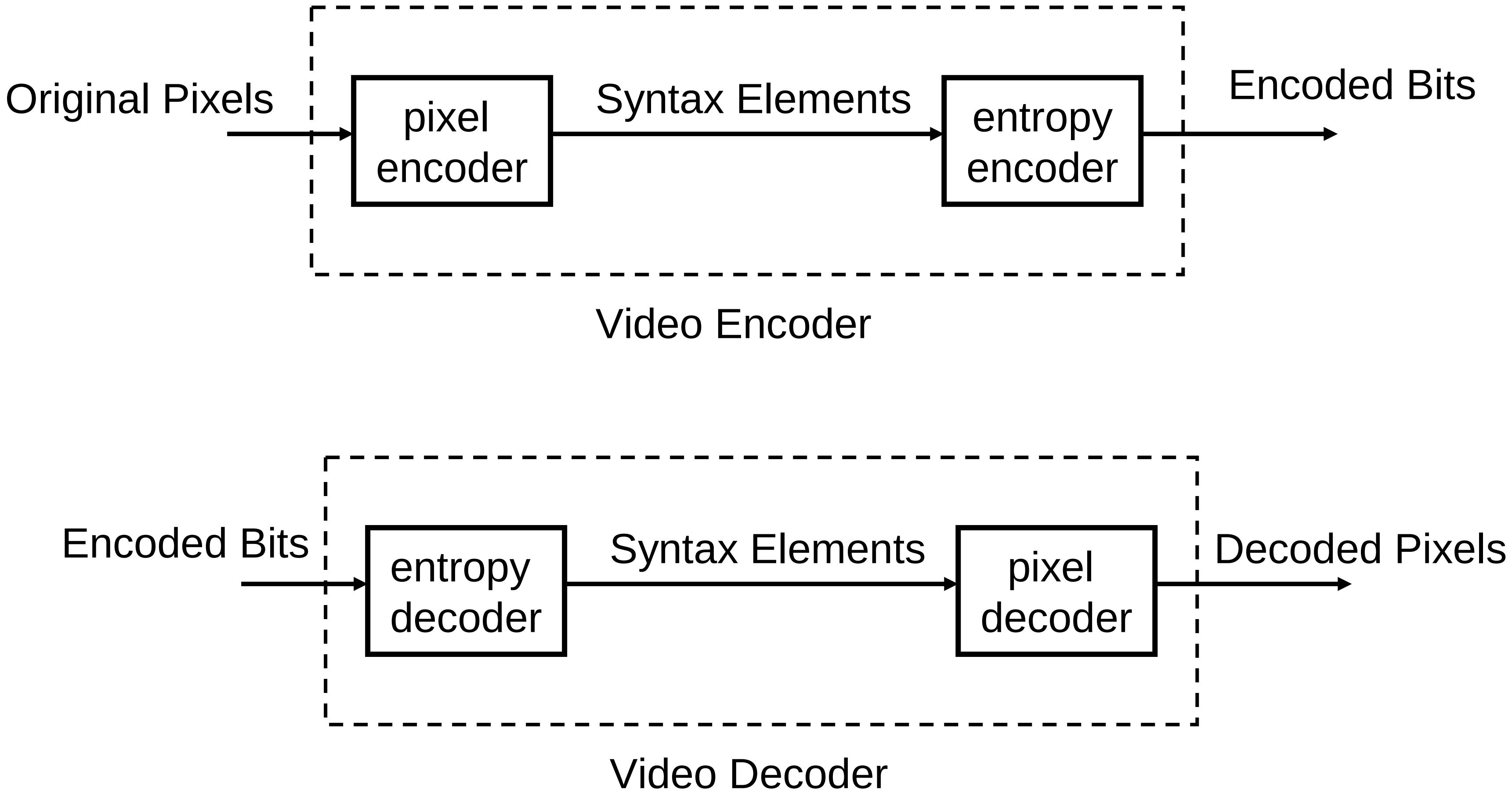
Joint Collaborative Team on Video Coding (JCT-VC)
of ITU-T SG16 WP3 and ISO/IEC JTb1/SC29/WG11
6th Meeting: Turin, IT, 13-22 July, 2011

Texas Instruments

Overview

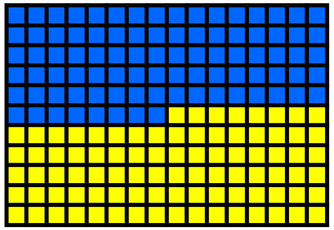
- Purpose of slices: Parallel Processing (and error resilience)
- Evaluate based on the following metrics
  - Coding efficiency
  - Throughput → Workload balancing
    - bit-rate workload (entropy)
    - pixel workload (prediction & reconstruction)
    - **For all approaches, to balance both requires either**
      - **frame buffer (latency and BW challenge)**
      - **restrictions to bits/bins and pixels simultaneously (RC challenge)**
  - Latency
  - Memory Size/Bandwidth (ME, last line, buffering)
  - Communications between cores
  - Single Core processing
- Placing a bin/bit limit will require serial processing at encoder

Encoder and Decoder

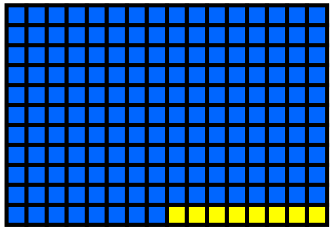


Key differences between Slice approaches

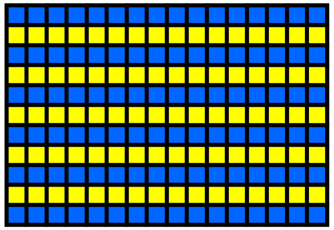
Allocation of pixel data to each slice



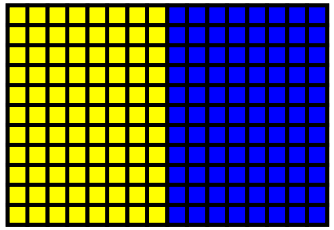
Regular Slices
&
Entropy Slices
(LCU limit)



Entropy Slices
(Bin Limit)



Interleaved Entropy
Slices (IES)
& Wavefront Parallel
Processing (WPP)



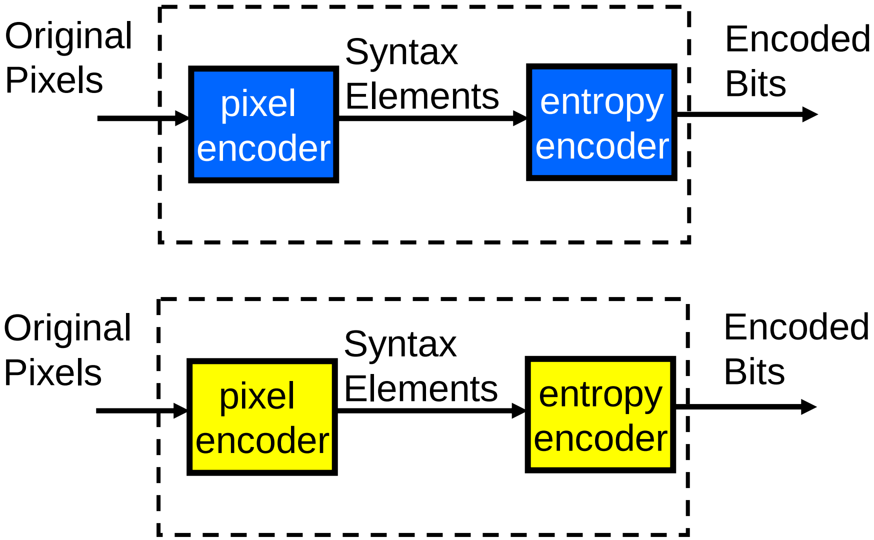
Tiles

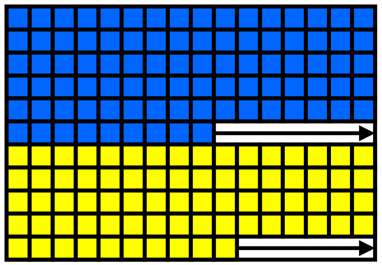
Dependencies between slices

	Entropy encode/decode	Pixel encode/decode
Regular Slice	No	No
Entropy Slice	No	Yes
IES/WPP	Yes	Yes
Tiles	No	No

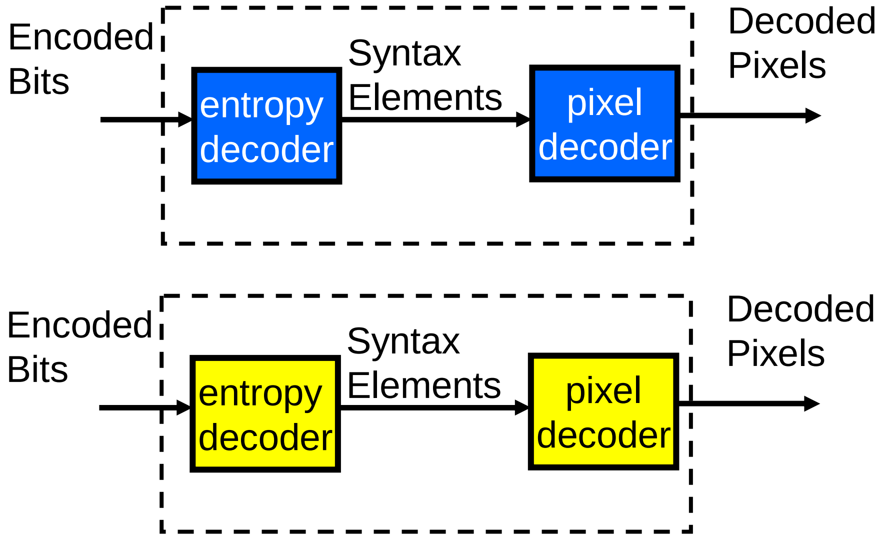
Regular Slices

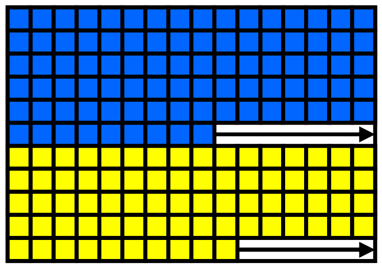
Encoding is fully parallel





Decoding is fully parallel



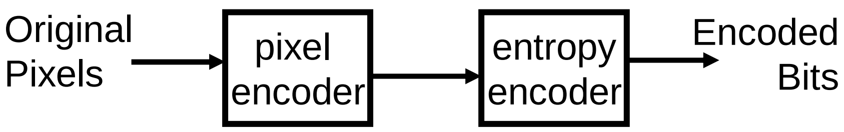


Regular Slices

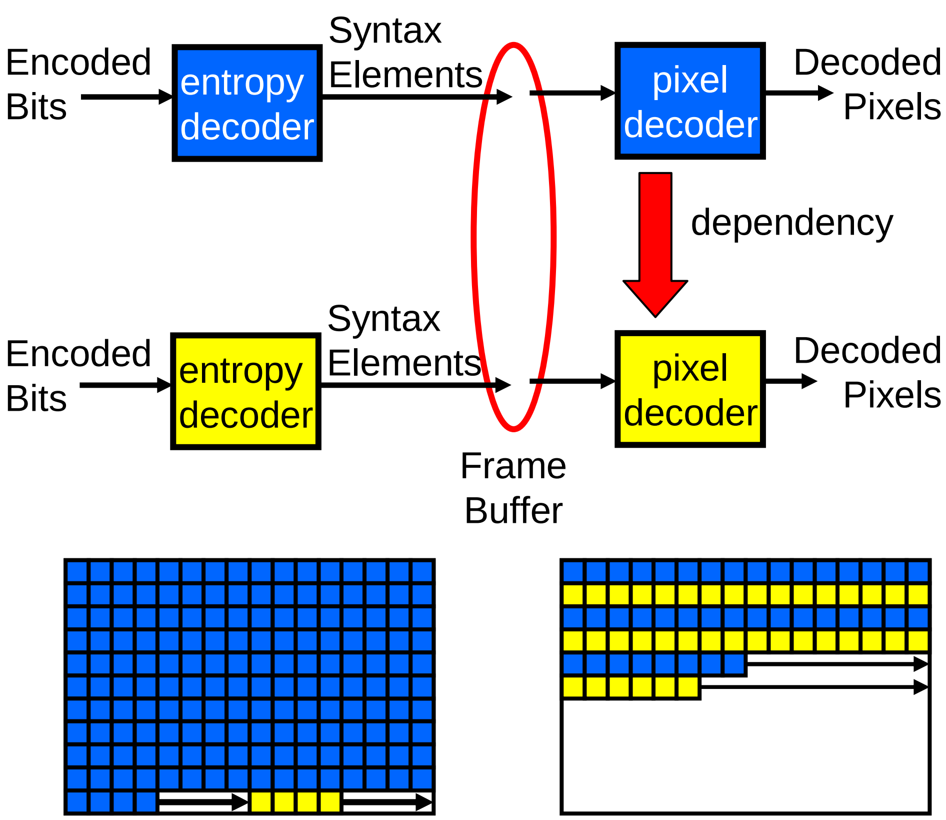
Coding Efficiency	High loss since no prediction or entropy dependencies across slices; slice header + pointer overhead 0.3 (AI), 0.8 (RA), 1.2 (LD)% 4 slices per frame in High Efficiency (equal pixels per slice)
Throughput	Either bits or pixels balanced
Latency	No change
Memory	No change
Communication between cores	None
Single Core	No Overhead

Entropy Slices (Bin Limit)

Entropy encoding must be performed serially



Decoding in parallel
frame buffering (required)



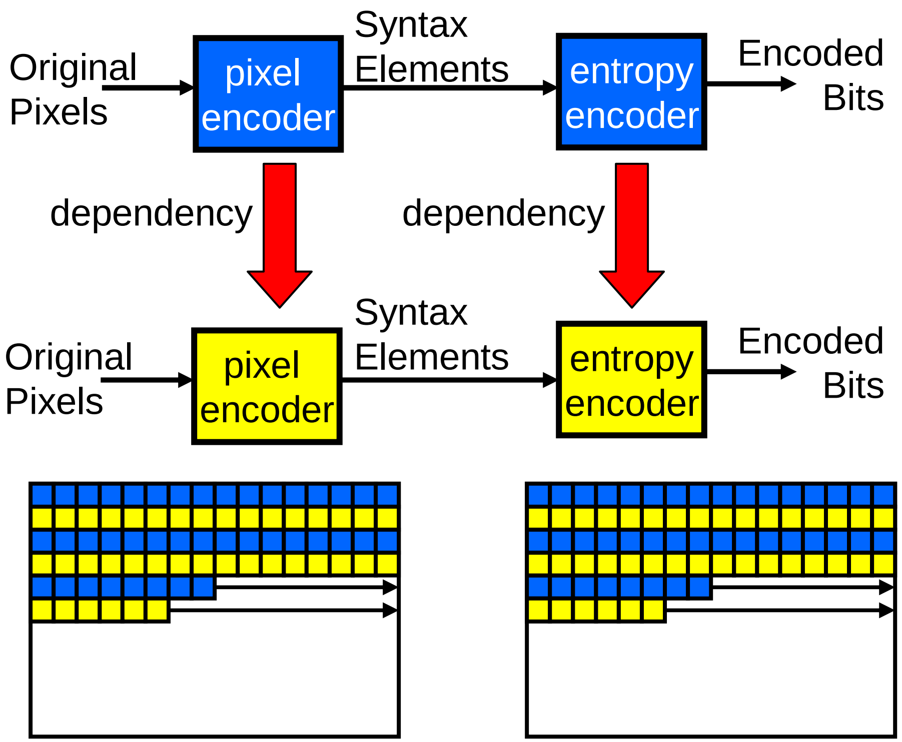
Entropy Slices

Coding Efficiency	Some loss since no entropy dependencies across slices (Intra prediction allowed across slices); reduced slice header, but pointer overhead remains 0.1 (AI), 0.5 (RA), 0.9 (LD)% [4 slices per frame in High Efficiency (equal pixels)]
Throughput	Upper bound on bins or upper bound on pixels; for bin limit, encoder serial*
Latency	Minimum 1 frame at decoder
Memory	**Frame buffer at decoder required**
Communication between cores	Only for pixel encoding/decoding. None for entropy encoding/decoding.
Single Core	Need to reinitialize CABAC context more frequently

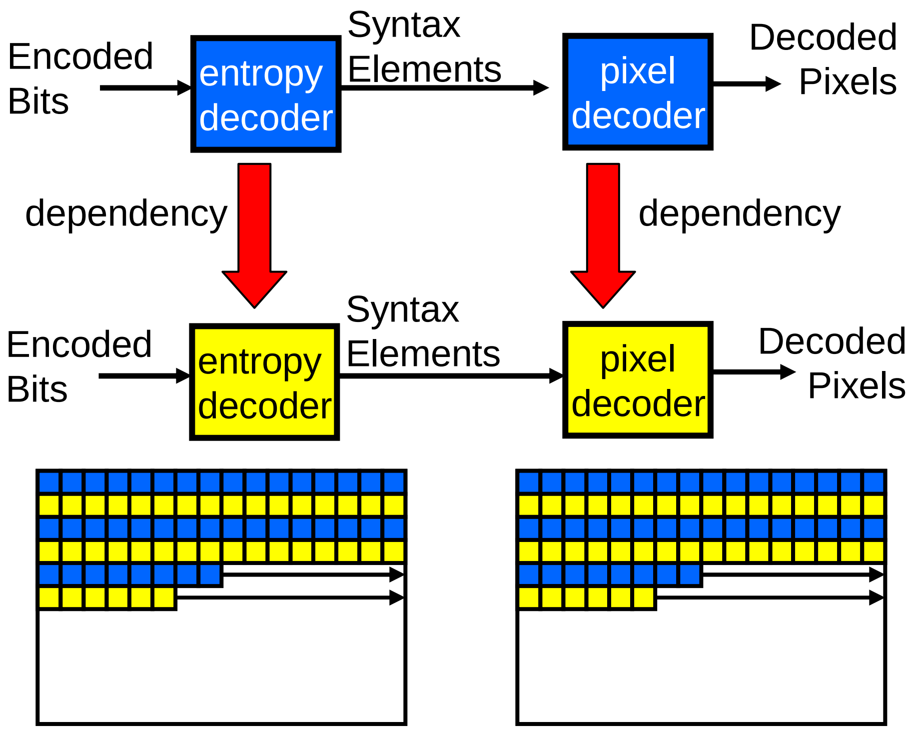
- Note: If LCU limit rather than bin limit, parallel processing possible at encoder as well, but requires frame decoupling at encoder.

IES/WPP

Encoding is fully parallel



Decoding is fully parallel



<u>Difference between IES and WPP</u>

IES – multiple rows are same slice, initialization of context not dependent on other rows

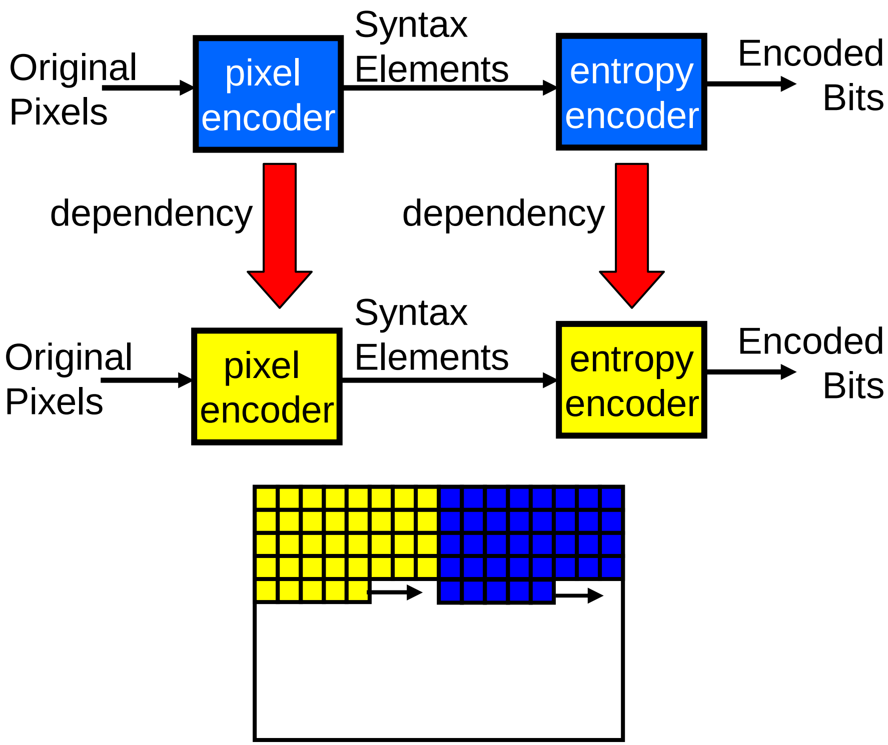
WPP – context initialization from top left neighbor (memory impact) and each row is one slice

IES/WPP

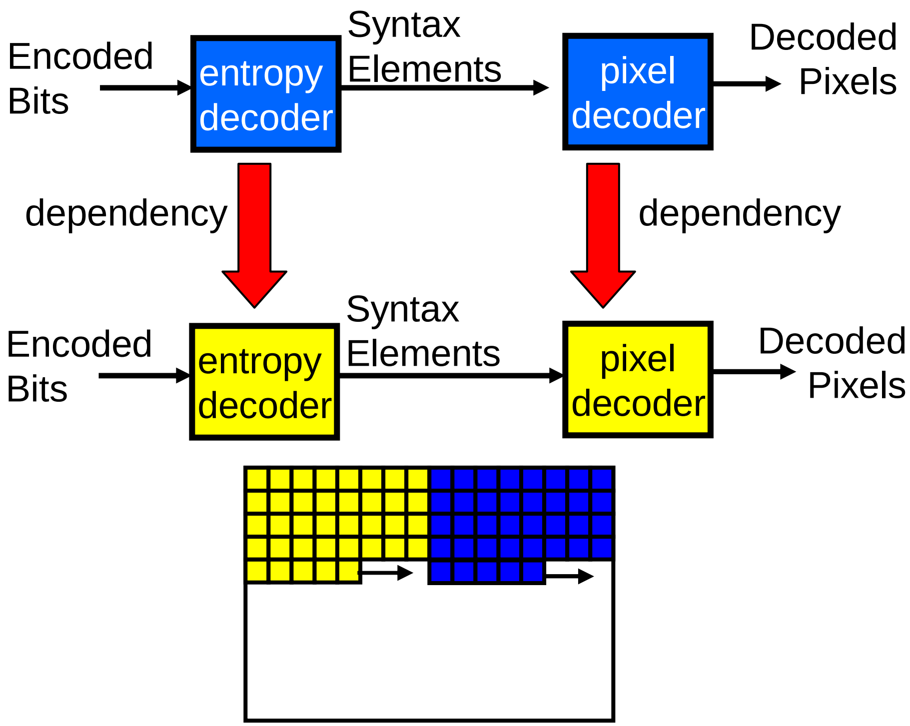
Coding Efficiency	Minimal coding loss; uses reduced slice header; loss mostly due to pointer overhead
Throughput	Average performance more balanced (still need frame buffer for worst case)
Latency	~N LCUs for N slices
Memory	Reduce line buffer bandwidth
Communication between cores	Yes (synchronize through FIFOs at an LCU level)
Single Core	Yes, need to store context memory and bitstream buffer for each slice

Tiles

Encoding is fully parallel



Decoding is fully parallel



Tiles

Coding Efficiency	High loss since no prediction or entropy dependencies across slices; slice header / pointer overhead
Throughput	Either pixels or bits balanced
Latency	Potentially faster encoding than other approaches by 1/N of LCU row; potential high delay for decoder if display is raster scan and single core (or switch between slices)
Memory	Reduce ME on-chip memory requirements (but not bandwidth); with same on-chip memory size, can have better coding efficiency (greater vertical support)
Communication between cores	No
Single Core	No overhead

Summary of Metrics vs. Serial

	Regular Slices	Entropy Slices	Interleaved ES/ WPP	Tiles
Coding Efficiency	Worst	Fair	Best	Fair; if account for ME, Best
Throughput *	Bits or pixels	Bins or pixels	Bins or pixels (on average more balanced)	Bits or pixels
Latency	No change	1 frame at decoder (also at encoder for LCU limit)	~N LCUs for N slices	Potentially faster at encoder
Memory	No change	Additional Frame buffering required	Reduce last line buffer and ME cache BW	Reduce ME cache size, or better coding efficiency
Communication between cores	No	No	Yes	No
Single Core	No change	Initialize CABAC more frequency	Additional storage of context memory and bitstream buffer	May have delay at decoder

*All approaches need frame buffer to address worst case for both bits/bins and pixels