



# Wavefront Parallel Processing with Tiles

Chih-Wei Hsu, Chia-Yang Tsai, Yu-Wen Huang, and Shawmin Lei



# Overall Summary

- Extend WPP (JCTVC-E196) to tiles (JCTVC-D227) for parallel processing
- The proposed WPP with tiles scheme:
  - In comparison with WPP, 80% causality checks can be reduced
  - In comparison with tiles, better coding efficiency is maintained under the same parallelism conditions

| BD-rates  | In Comparison with Tiles | In Comparison with WPP |
|-----------|--------------------------|------------------------|
| 2 threads | -0.5%                    | -0.6%                  |
| 3 threads | -0.8%                    | -0.5%                  |
| 4 threads | -1.2%                    | -0.5%                  |

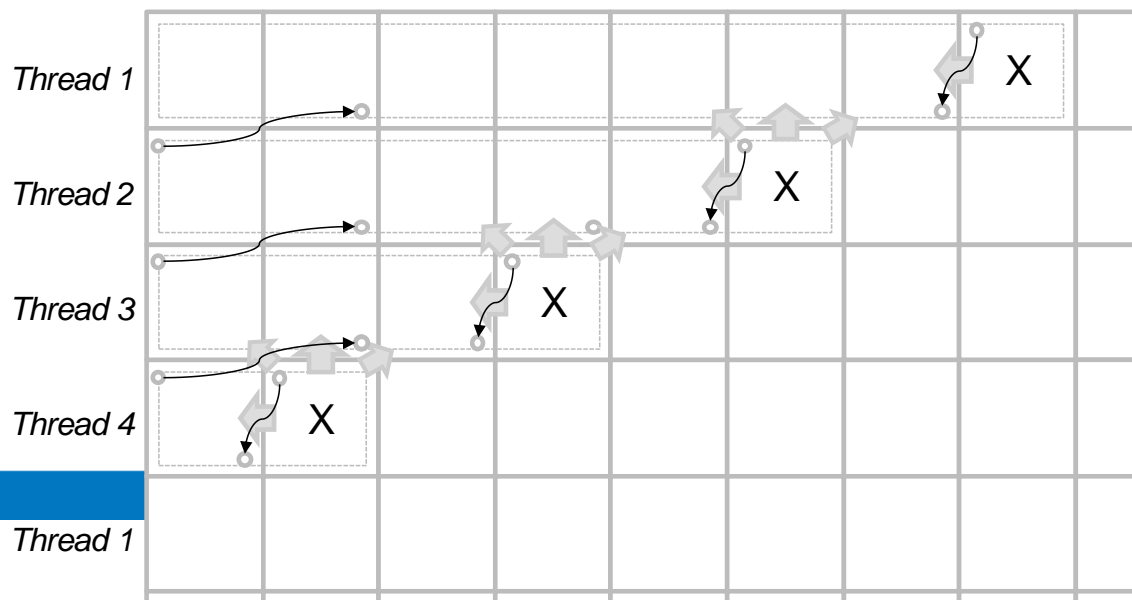
# Outline

- Introduction
- Proposed methods
- Simulations results
- Conclusions

# Wavefront Parallel Processing [JCTVC-E196]

- Wavefront processing order maintains spatial dependency
- Quick learning of CABAC probabilities
  - The first LCU of each row is initialized with the probabilities obtained after the second LCU of the upper row is processed
- WPP causality check:**
  - Check if the upper right LCU has been processed
- For multi-core systems to perform WPP
  - Takes more interactions among cores to preserve the correct processing order between threads

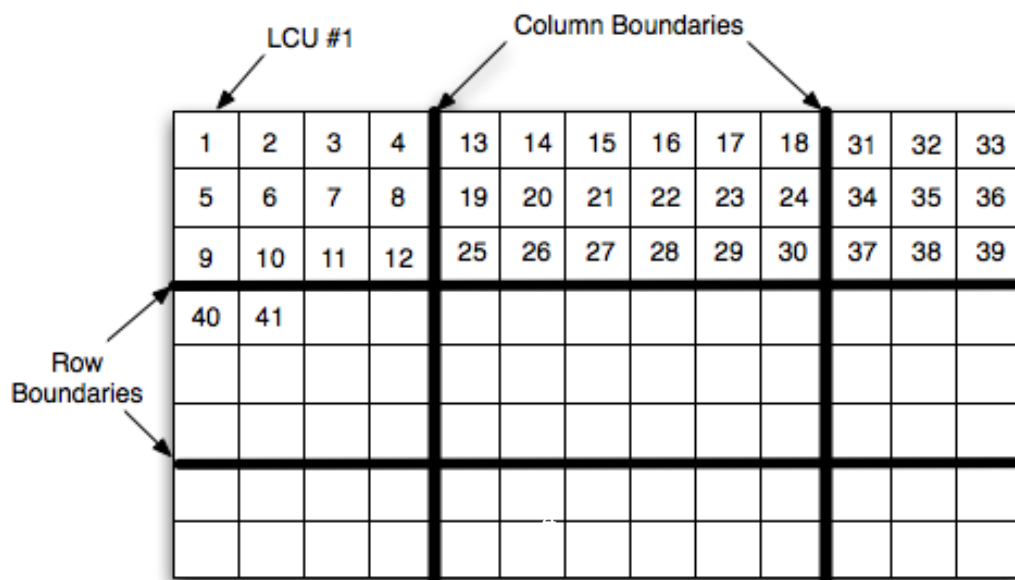
## WPP with 4 threads



# Tiles [JCTVC-D227, E408]

- Partition picture into rectangular segments for parallel processing
  - LCUs are processed in raster scan order within each tile
  - Tiles are processed in raster scan order within the picture
- Row and column boundaries break dependency as slice
  - Suffer coding efficiency loss

## Tiles partitioning using 3 columns and 3 rows

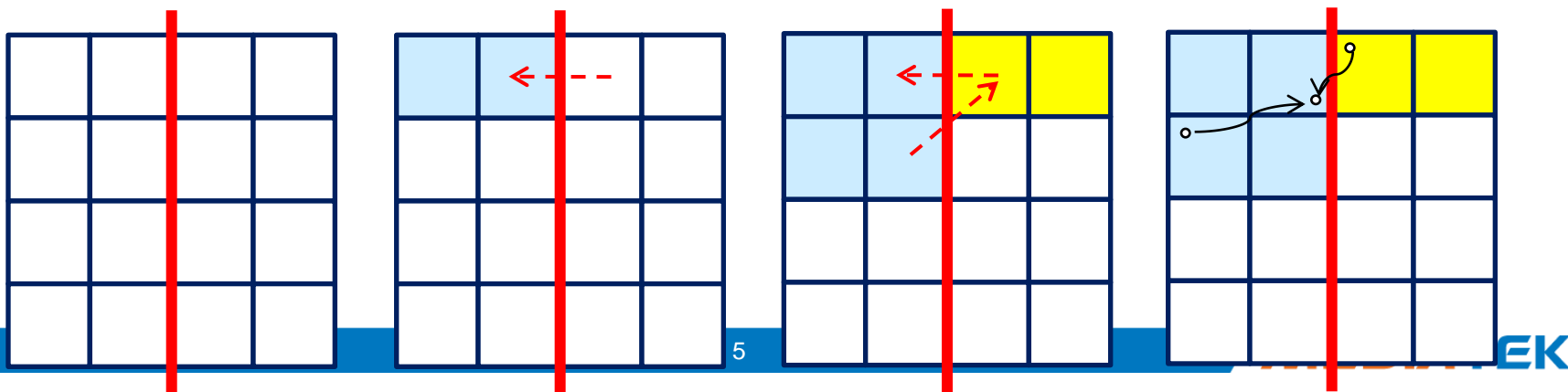


# Proposed Method: WPP with Tiles

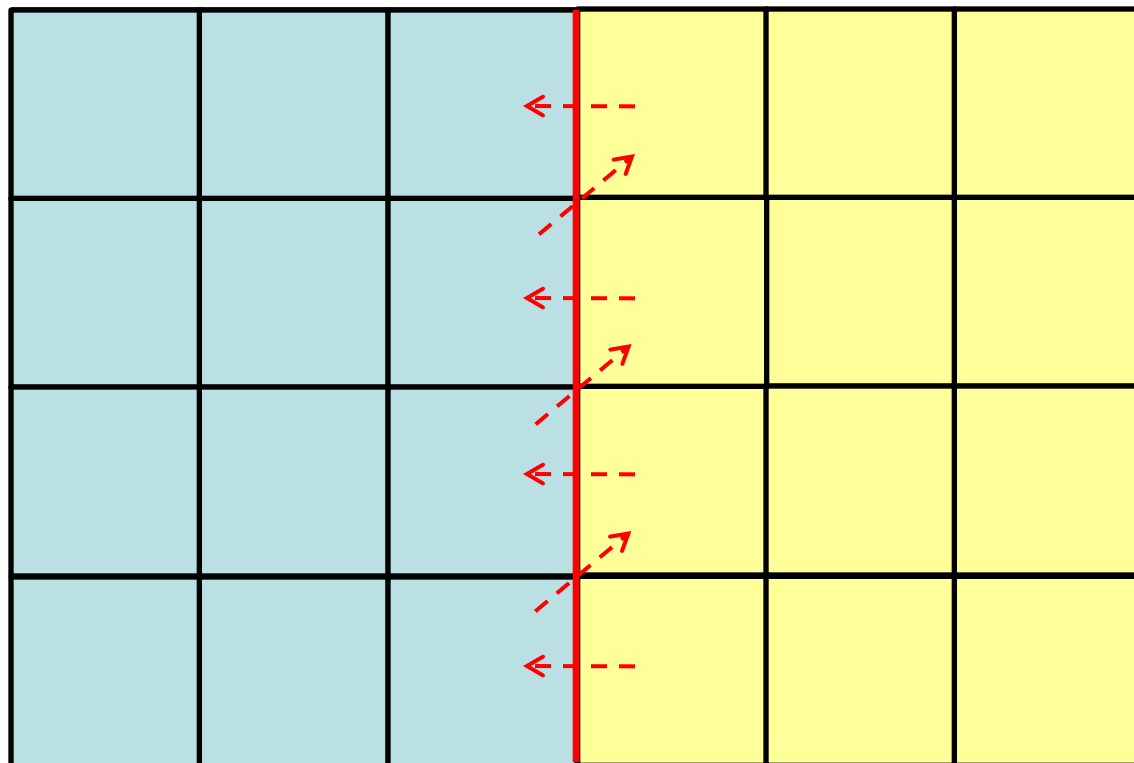
- WPP concept is extended and applied to tiles with columns of LCUs
- Causality checks only happen on the LCUs at tile boundaries
  - For tile boundary LCU at right tile: check left LCU
  - For tile boundary LCU at left tile: check upper-right LCU
- CABAC probabilities inheritance
  - Inner tile: raster-scan order
  - Inter tile: only the first LCU row; left to right

## WPP with 2 tiles

**<- - - - : Causality check**



# WPP with 2 Tiles

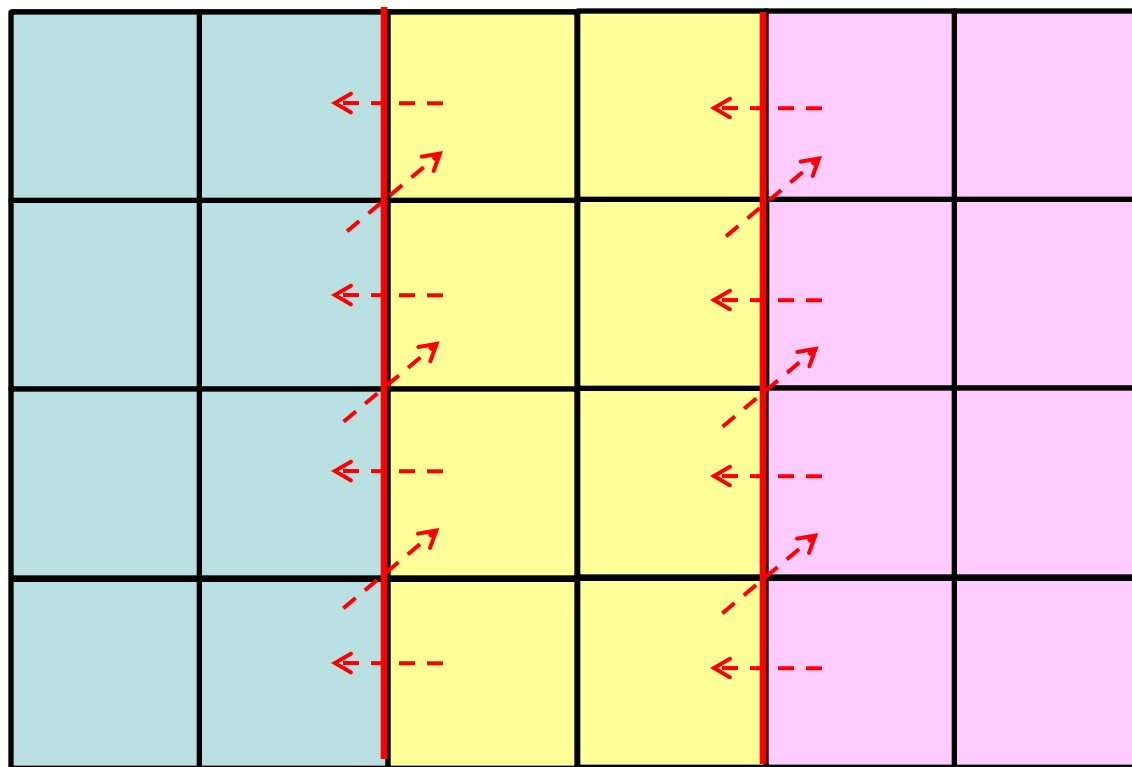


← --- Causality check

Tile boundary

# WPP with 3 Tiles

← --- Causality check



Tile boundary

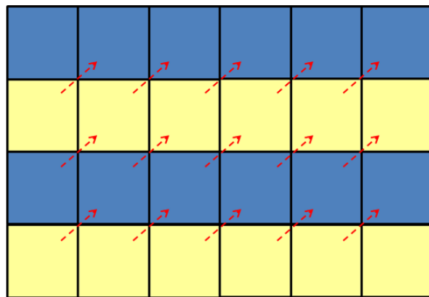
Tile boundary



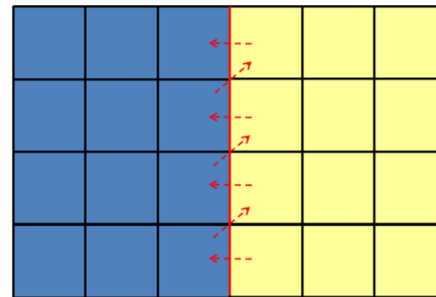
# Number of Causality Checks (1/2)

- WPP
  - Depends on number of LCUs of entire picture
  - **5x3=15** in the 2-thread example
- The proposed “WPP with tiles”
  - Depends on number of LCUs of picture height & number of tiles
  - **(3+4)x1=7** in the 2-thread example

2-thread WPP



2-thread “WPP with tiles”



 : Causality check

# Number of Causality Checks (2/2)

- 4-thread case, in comparison with WPP,
  - Class A: 16% number of causality checks
  - Class B: 22% number of causality checks
  - Class E: 33% number of causality checks
- When the picture size increases, the reduction is more significant

|         | WPP     | Proposed “WPP with tiles” |             |             |
|---------|---------|---------------------------|-------------|-------------|
| Threads | 2, 3, 4 | 2                         | 3           | 4           |
| Class A | 936     | 49 (5.2%)                 | 98 (10.5%)  | 147 (15.7%) |
| Class B | 464     | 33 (7.1%)                 | 66 (14.22%) | 99 (21.3%)  |
| Class E | 209     | 23 (11.0%)                | 46 (22.0%)  | 69 (33.0%)  |

# Simulation Results: 2 Threads

- Anchor
  - HM-3.0
  - High-efficiency configurations in JCTVC-E700
- Results
  - The proposed scheme is 0.5% better than tiles
  - The proposed scheme is 0.6% better than WPP

| BD-rate                      | HE-All Intra | HE-Random Access | HE-Low Delay |
|------------------------------|--------------|------------------|--------------|
| Tiles                        | 0.4%         | 0.6%             | 0.9%         |
| WPP                          | 0.2%         | 0.7%             | 1.2%         |
| WPP with tiles<br>(proposed) | 0%           | 0.1%             | 0.3%         |

# Simulation Results: 3 Threads

- Anchor
  - HM-3.0
  - High-efficiency configurations in JCTVC-E700
- Results
  - The proposed scheme is 0.8% better than tiles
  - The proposed scheme is 0.5% better than WPP

| BD-rate                      | HE-All Intra | HE-Random Access | HE-Low Delay |
|------------------------------|--------------|------------------|--------------|
| Tiles                        | 0.7%         | 1.0%             | 1.5%         |
| WPP                          | 0.2%         | 0.8%             | 1.3%         |
| WPP with tiles<br>(proposed) | 0%           | 0.2%             | 0.6%         |

# Simulation Results: 4 Threads

- Anchor
  - HM-3.0
  - High-efficiency configurations in JCTVC-E700
- Results
  - The proposed scheme is 1.2% better than tiles
  - The proposed scheme is 0.5% better than WPP

| BD-rate                      | HE-All Intra | HE-Random Access | HE-Low Delay |
|------------------------------|--------------|------------------|--------------|
| Tiles                        | 1.0%         | 1.7%             | 2.2%         |
| WPP                          | 0.2%         | 0.9%             | 1.5%         |
| WPP with tiles<br>(proposed) | 0%           | 0.4%             | 0.8%         |

# Crosscheck Verifications

- We thank Cisco and TI for crosschecking our proposal
  - JCTVC-F450
- The BD-rate results are confirmed

# Conclusions

- In this contribution, WPP concept was extended and applied to tiles for parallel processing
- The proposed method was compared with both WPP and tiles
  - 0.5% and 0.6% better than tiles and WPP respectively for 2 threads
  - 0.8% and 0.5% better than tiles and WPP respectively for 3 threads
  - 1.2% and 0.5% better than tiles and WPP respectively for 4 threads
  - The number of causality checks of the proposed WPP with tiles is significantly less than that of WPP only
    - < 16% for class A (2560x1600)
    - < 22% for class B (1920x1080)
    - < 33% for class E (1280x720)