

# NON-CE2: MODIFIED CODING ORDER OF SYNTAX ELEMENTS IN PALETTE MODE


Weijia Zhu, Li Zhang, Jizheng Xu and Hsiao-Chiang Chuang




# Background

- In the current CG based palette mode coding,
  - *Three stages (i.e., all copy flags and run types; palette indices; and escape values) are required to be performed for a CG coding process*


palette_coding( x0, y0, cbWidth, cbHeight, treeType ) {	Descriptor
...	
for( subSetId = 0; subSetId <= ( cbWidth * cbHeight - 1 ) / 16; subSetId++ ) {	
...	
while( PaletteScanPos < maxSubPos ) {	
...	
run_copy_flag	ac(v)
...	
copy_above_palette_indices_flag	ac(v)
...	
PaletteScanPos ++	
}	
PaletteScanPos = minSubPos	
while( PaletteScanPos < maxSubPos ) {	
...	
palette_idx_idc	ac(v)
...	
PaletteScanPos ++	
}	
if( palette_escape_val_present_flag ) {	
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )	
for( sPos = minSubPos; sPos < maxSubPos; sPos++ ) {	
...	
palette_escape_val	ac(v)
...	
}	
}	



1<sup>st</sup> loop



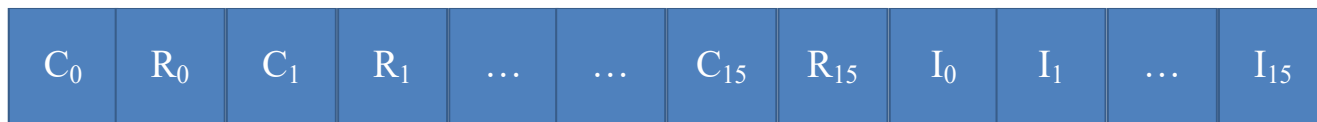
2<sup>nd</sup> loop



3<sup>rd</sup> loop

# Motivation

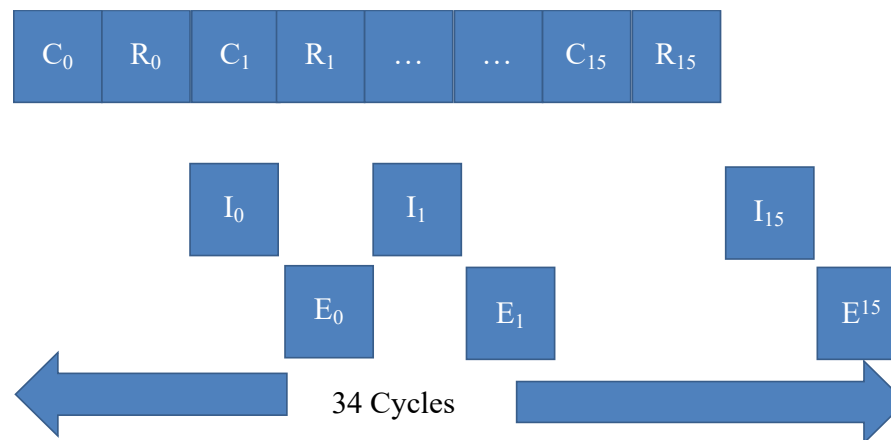
- Decoding a CG needs 49 cycles, which can be reduced by a pixel-by-pixel decoding way.



- The reconstruction of a pixel in a CG has to wait for the decoding of whole CG, definitely increasing the latency.

# Proposed Method

- Proposes to code the syntax elements in a pixel-by-pixel way in a CG.
  - *The copy flag, run type, index and escape value of every pixel are coded before those associated with a next pixel.*



# Text changes

	Descriptor
palette_coding( x0, y0, cbWidth, cbHeight, treeType ) {	
...	
for( subSetId = 0; subSetId <= ( cbWidth * cbHeight - 1 ) / 16; subSetId++ ) {	
...	
while( PaletteScanPos < maxSubPos ) {	
...	
if ( MaxPaletteIndex > 0 && PaletteScanPos > 0 ) {	
<b>run_copy_flag</b>	ae(v)
RunCopyMap[ xC ][ yC ] = run_copy_flag	
}	
CopyAboveIndicesFlag[ xC ][ yC ] = 0	
if( MaxPaletteIndex > 0 && !RunCopyMap[ xC ][ yC ] ) {	
if( ( ( !palette_transpose_flag && yC > 0 )   ( palette_transpose_flag && xC > 0 ) )	
&& CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {	
<b>copy above palette indices flag</b>	ae(v)
CopyAboveIndicesFlag[ xC ][ yC ] = copy above palette indices flag	
}	
...	
} else	
...	
<del>    PaletteScanPos ++</del>	
<del>    }</del>	
<del>    PaletteScanPos = minSubPos</del>	
<del>    while( PaletteScanPos &lt; maxSubPos ) {</del>	
<del>        xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]</del>	
<del>        yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]</del>	
<del>        if( PaletteScanPos &gt; 0 ) {</del>	
<del>            xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]</del>	
<del>            ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]</del>	
<del>        }</del>	
if( MaxPaletteIndex > 0 )	
if ( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
if( MaxPaletteIndex - adjust > 0 )	
<b>palette_idx_idc</b>	ae(v)
adjust = 1	
}	
..	
<del>    PaletteScanPos ++</del>	
<del>    }</del>	
if( palette_escape_val_present_flag ) {	
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )	
<del>for( sPos = minSubPos; sPos &lt; maxSubPos; sPos++ ) {</del>	
<del>    xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 0 ]</del>	
<del>    yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 1 ]</del>	
if( PaletteIndexMap[ cIdx ][ xC ][ yC ] == MaxPaletteIndex ) {	
<b>palette_escape_val</b>	ae(v)
PaletteEscapeVal[ cIdx ][ xC ][ yC ] = palette_escape_val	
}	
}	
}	
}	

# Benefits

- Only needs 34 cycles to decode a CG
  - *30% of number of cycles reduction*
- Every palette coded sample can be directly reconstructed without waiting for the whole CG

# Results

- Anchor: VTM7.0
- Test: JVET-Q0477
- Dual Tree on

	All Intra					Random Access					Low Delay B				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
TGM 1080p	0.00%	0.00%	0.00%	99%	99%	0.00%	0.00%	0.00%	98%	98%	0.00%	0.00%	0.00%	101%	98%
TGM 720p	0.00%	0.00%	0.00%	98%	93%	0.00%	0.00%	0.00%	99%	104%	0.00%	0.00%	0.00%	101%	98%
Animation	0.00%	0.00%	0.00%	97%	96%	0.00%	0.00%	0.00%	98%	98%	0.00%	0.00%	0.00%	99%	100%
Mixed content	0.00%	0.00%	0.00%	98%	96%	0.00%	0.00%	0.00%	98%	100%	0.00%	0.00%	0.00%	96%	98%
Camera-Captured content	0.00%	0.00%	0.00%	100%	100%	0.00%	0.00%	0.00%	98%	97%	0.00%	0.00%	0.00%	100%	100%
<b>Overall</b>	0.00%	0.00%	0.00%	98%	97%	0.00%	0.00%	0.00%	98%	100%	0.00%	0.00%	0.00%	99%	99%

# Results

- Anchor: VTM7.0
- Test: JVET-Q0477
- Dual Tree off

	All Intra					Random Access					Low Delay B				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
TGM 1080p	0.00%	0.00%	0.00%	98%	98%	0.00%	0.00%	0.00%	99%	99%	0.00%	0.00%	0.00%	98%	100%
TGM 720p	0.00%	0.00%	0.00%	98%	96%	0.00%	0.00%	0.00%	98%	96%	0.00%	0.00%	0.00%	98%	95%
Animation	0.00%	0.00%	0.00%	98%	99%	0.00%	0.00%	0.00%	99%	98%	0.00%	0.00%	0.00%	99%	97%
Mixed content	0.00%	0.00%	0.00%	97%	97%	0.00%	0.00%	0.00%	98%	98%	0.00%	0.00%	0.00%	98%	100%
Camera-Captured content	0.00%	0.00%	0.00%	98%	98%	0.00%	0.00%	0.00%	98%	98%	0.00%	0.00%	0.00%	98%	99%
<b>Overall</b>	0.00%	0.00%	0.00%	98%	98%	0.00%	0.00%	0.00%	98%	98%	0.00%	0.00%	0.00%	98%	98%