



<10th JCT-VC meeting @Stockholm, July 2012>

◁JCTVC-J0202/J0212/J0313> ScanType Swapping for Transform Skip Mode

Robert A. Cohen (MERL)

Hui Yong Kim (ETRI)

Toshiyasu Sugio (Panasonic)



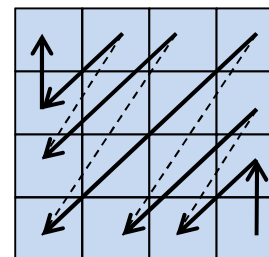
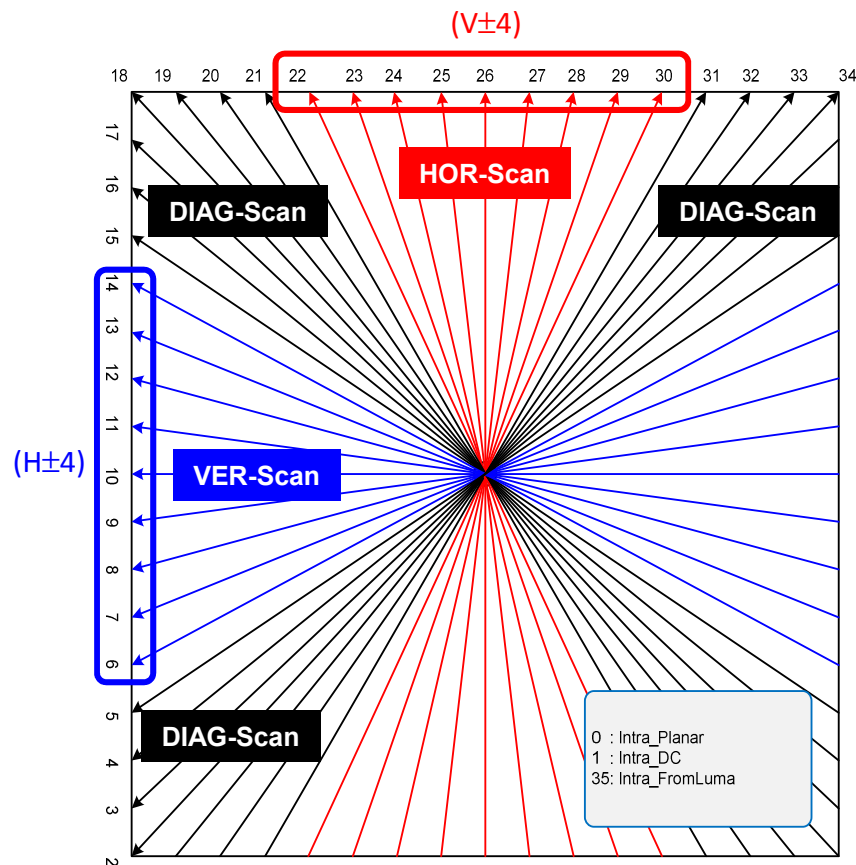
2012-07-13



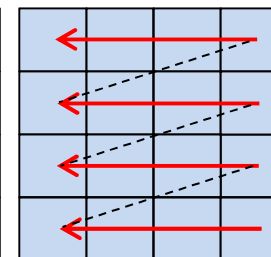
Introduction

❑ HM 7.0 Mode-Dependent Coefficient Scanning

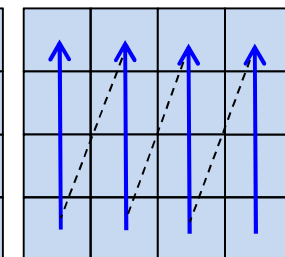
- ❖ Scanning direction is roughly perpendicular to prediction direction
- ❖ *Make sense for transformed residue, but not for spatial residue*



DIAG-Scan (0)



HOR-Scan (1)

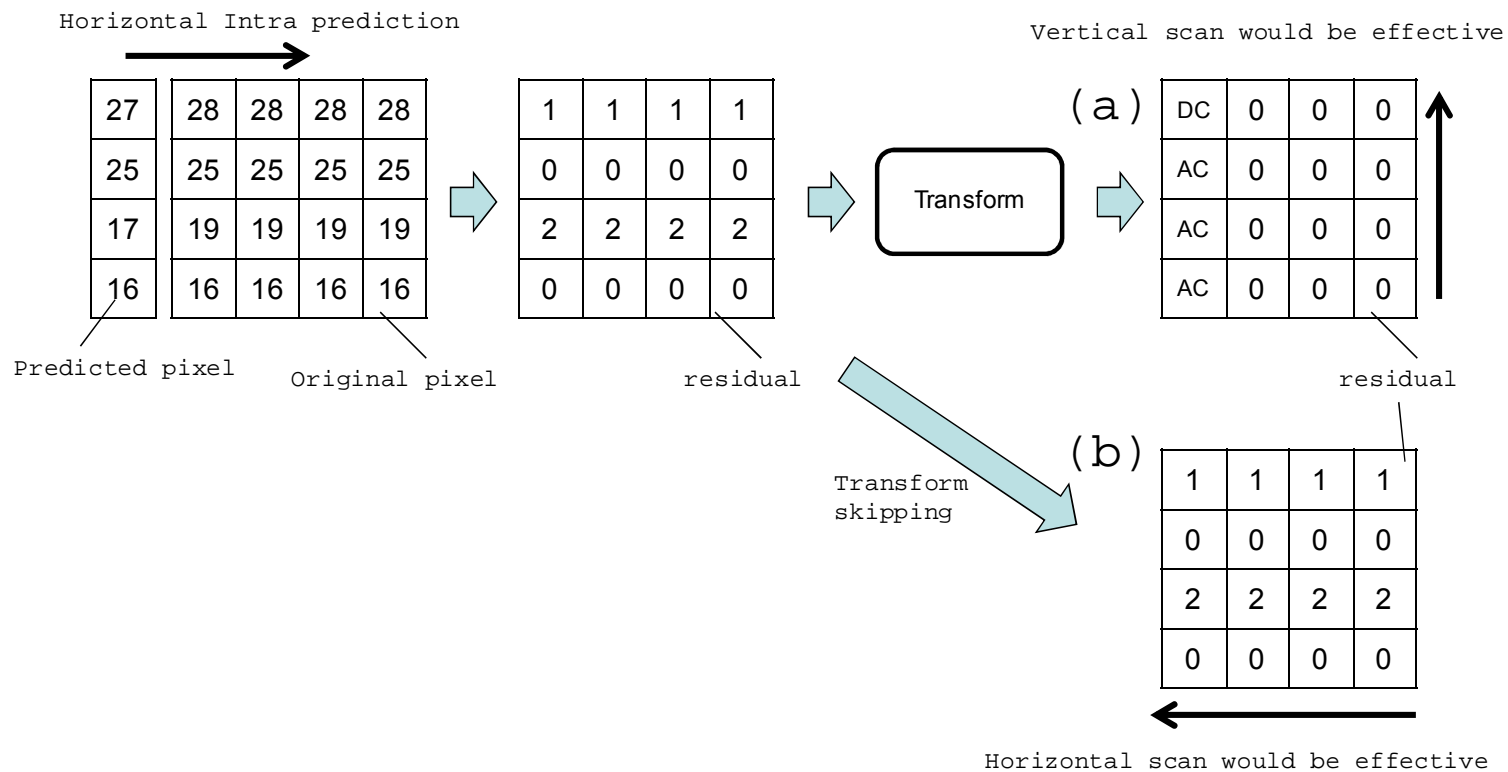


VER-Scan (2)

A Case Study

❑ Horizontal intra prediction case

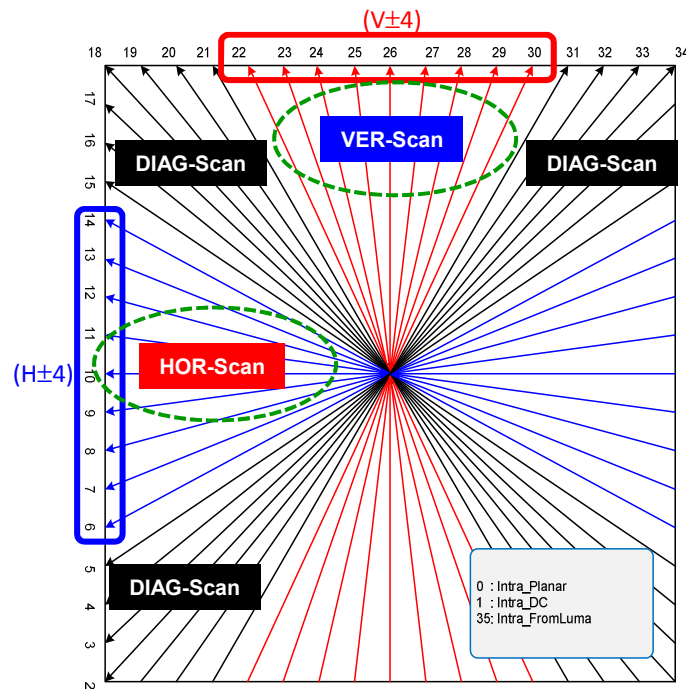
- ❖ Vertical scan is preferable for transformed residue since zero coefficients tend to be successive in a vertical direction after transforming
- ❖ However, if the transform were skipped, the vertical scan would not be effective since zero coefficients tend to be successive in horizontal direction



Proposal

❑ ScanType swapping for transform skipped block

- ❖ Use vertical scan for vertical prediction modes and horizontal scan for horizontal prediction modes
- ❖ Requires just two lines of syntax change. No change in decoding process needed.



	Descriptor
<code>residual_coding(x0, y0, log2TrafoWidth, log2TrafoHeight, scanIdx, cIdx) {</code>	
<code>if(log2TrafoWidth == 1 log2TrafoHeight == 1) {</code>	
<code>log2TrafoWidth = 2</code>	
<code>log2TrafoHeight = 2</code>	
<code>}</code>	
<code>if(transform_skip_enabled_flag && !cu_transquant_bypass_flag &&</code>	
<code>(PredMode == MODE_INTRA) &&</code>	
<code>(log2TrafoWidth == 2) && (log2TrafoHeight == 2))</code>	
<code>transform_skip_flag[x0][y0][cIdx]</code>	ae(v)
<code>if(transform_skip_flag[x0][y0][cIdx])</code>	
<code>scanIdx = (scanIdx == 1) ? 2 : ((scanIdx == 2) ? 1 : 0)</code>	
<code>last_significant_coeff_x_prefix</code>	ae(v)
<code>last_significant_coeff_y_prefix</code>	ae(v)
<code>if(last_significant_coeff_x_prefix > 3)</code>	
<code>last_significant_coeff_x_suffix</code>	ae(v)
<code>if(last_significant_coeff_y_prefix > 3)</code>	
<code>last_significant_coeff_y_suffix</code>	ae(v)
<code>...</code>	
<code>}</code>	

Results

❑ Experimental Results

- ❖ Anchor: TS-enabled HM7.0
- ❖ JCTVC-J0202 is verified by Sharp (JCTVC-J0386)
- ❖ JCTVC-J0212 is verified by TI (JCTVC-J0212)
- ❖ JCTVC-J0313 is verified by I2R (JCTVC-J0407)

	All Intra Main			All Intra HE10		
	Y	U	V	Y	U	V
Class A	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class B	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%
Class C	-0.1%	-0.2%	-0.1%	-0.1%	-0.1%	-0.1%
Class D	-0.1%	-0.1%	-0.2%	-0.1%	-0.1%	-0.2%
Class E	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Overall	0.0%	-0.1%	-0.1%	0.0%	-0.1%	-0.1%
	0.0%	-0.1%	-0.1%	0.0%	-0.1%	-0.1%
Class F	-0.9%	-0.9%	-1.0%	-0.9%	-0.8%	-0.8%
Enc Time[%]	101%			100%		
Dec Time[%]	100%			99%		

	Random Access Main			Random Access HE10		
	Y	U	V	Y	U	V
Class A	0.0%	0.1%	-0.2%	0.0%	0.0%	-0.4%
Class B	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%
Class C	0.0%	-0.1%	-0.2%	0.0%	0.0%	-0.1%
Class D	-0.1%	-0.2%	-0.1%	0.0%	-0.2%	-0.6%
Class E						
Overall	0.0%	-0.1%	-0.1%	0.0%	-0.1%	-0.3%
	0.0%	-0.1%	-0.1%	0.0%	-0.1%	-0.3%
Class F	-0.7%	-0.6%	-0.9%	-0.7%	-0.5%	-0.6%
Enc Time[%]	100%			99%		
Dec Time[%]	100%			99%		

	Low delay B Main			Low delay B HE10		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	0.1%	-0.1%	0.0%	0.1%	-0.1%
Class C	0.0%	0.1%	-0.1%	0.0%	0.0%	0.0%
Class D	0.0%	0.4%	0.4%	0.0%	0.5%	0.0%
Class E	0.1%	-0.7%	-0.8%	0.1%	-0.9%	0.0%
Overall	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%
	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%
Class F	-0.5%	-0.3%	-0.9%	-0.3%	0.3%	0.3%
Enc Time[%]	100%			99%		
Dec Time[%]	100%			99%		

Conclusion

❑ ScanType swapping for transform skipped blocks

- ❖ 0.9% gain in class F with a simple index swapping with transform skip
- ❖ No change in decoding process involved
- ❖ Proponents suggest adopting the proposed method into the DIS

Changes to residual_coding():

...	
if(transform_skip_enabled_flag && !cu_transquant_bypass_flag && (PredMode == MODE_INTRA) && (log2TrafoWidth == 2) && (log2TrafoHeight == 2))	
transform_skip_flag[x0][y0][cIdx]	ae(v)
if(transform_skip_flag[x0][y0][cIdx])	
scanIdx = (scanIdx == 1) ? 2 : ((scanIdx == 2) ? 1 : 0)	
last_significant_coeff_x_prefix	ae(v)
last_significant_coeff_y_prefix	ae(v)
...	



Thank You Very Much !