

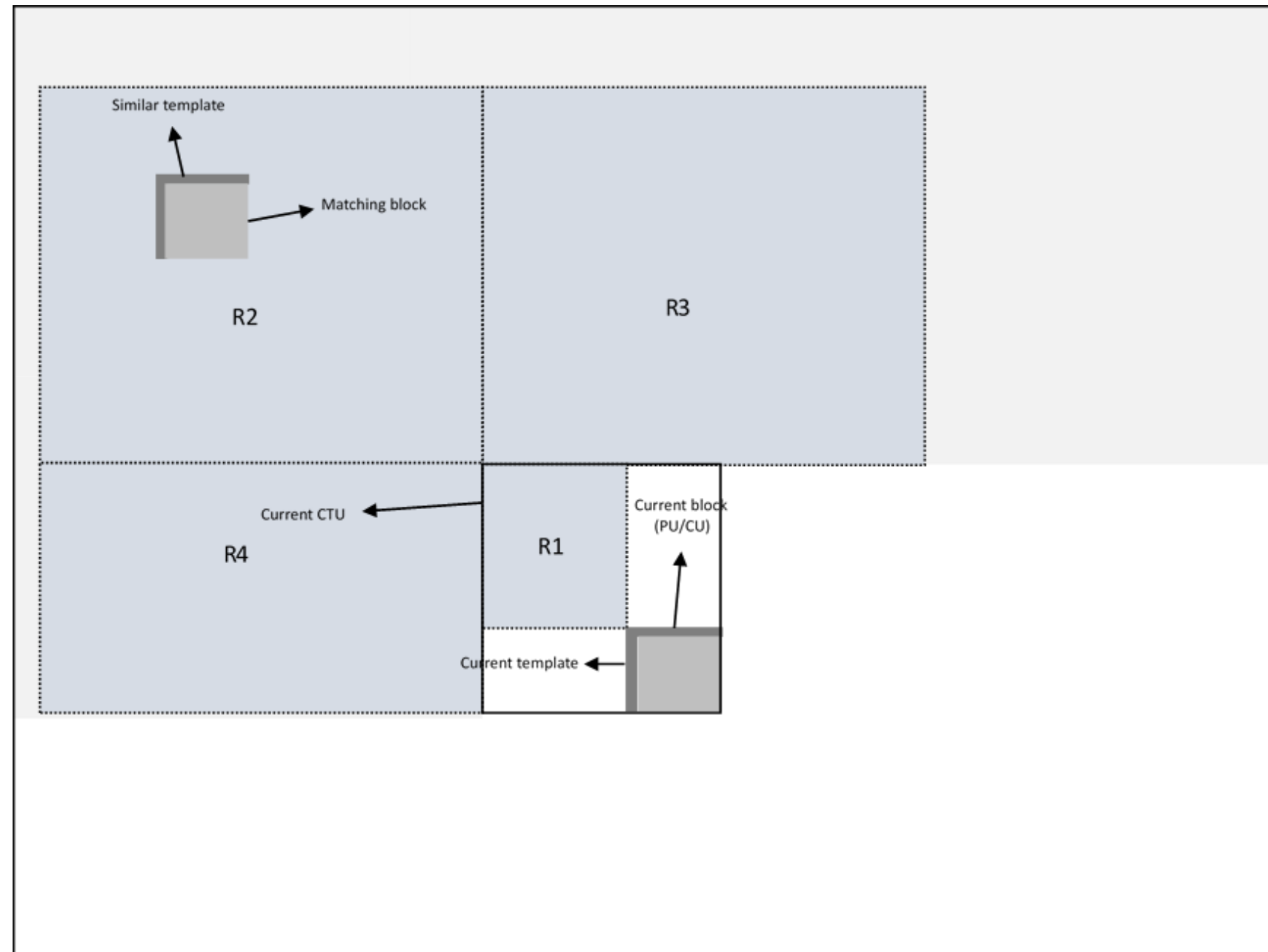
# JVET-X0124 AHG12: On signaling of intra template matching

Z. XIE, Y. YU, H. YU, L. XU, F. WANG, D. WANG

# Introduction

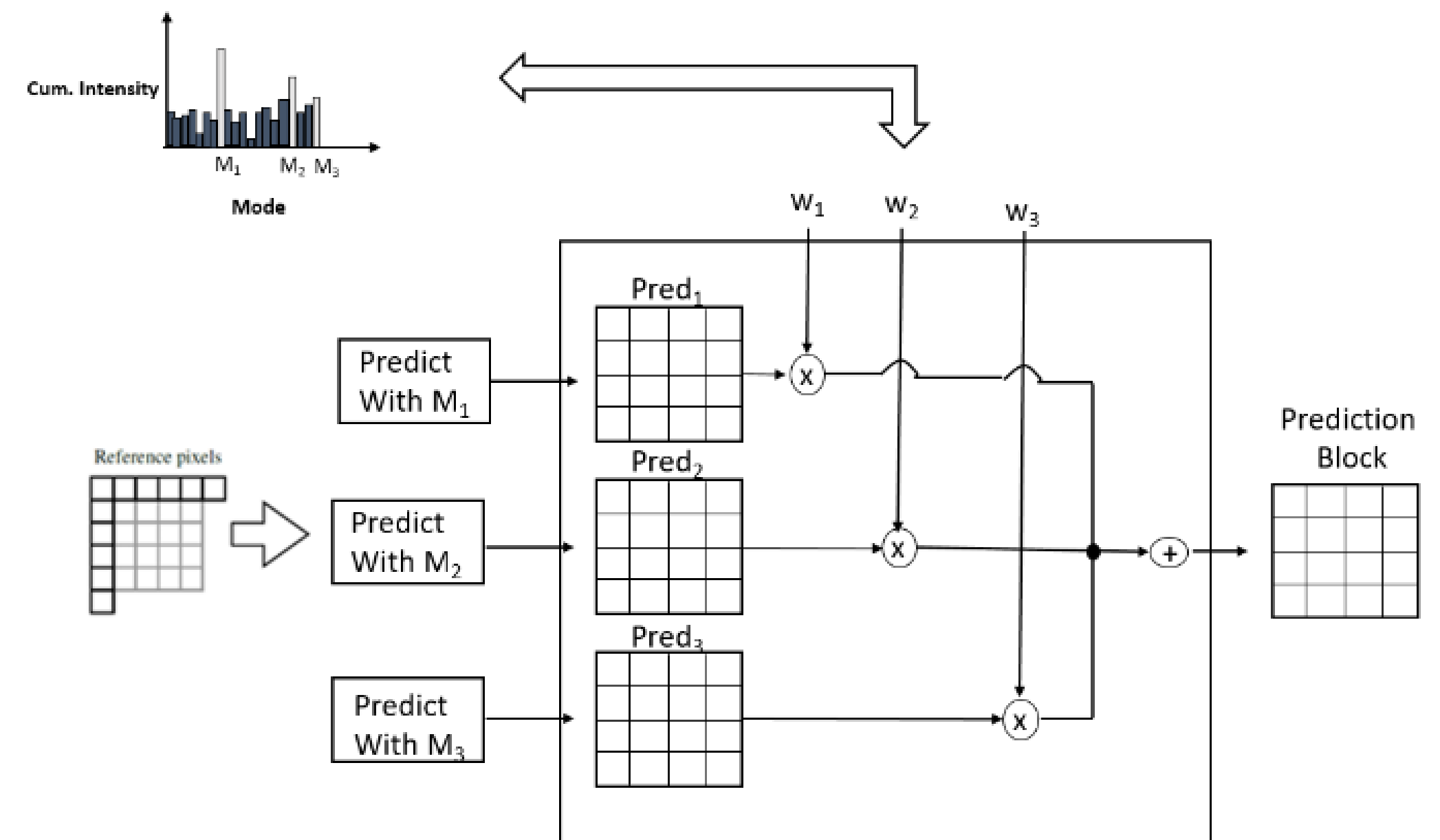
## Intra Template Matching Prediction (TMP)

- TMP is a special intra prediction mode that copies the best prediction block from the reconstructed part of the current frame, whose L-shaped template matches the current template.
- The usage of this tool is signaled at CU level through a dedicated flag.



## Decoder-side Intra Mode Derivation (DIMD)

- DIMD is an intra tool that derives two intra modes from the reconstructed neighboring samples based on Histogram of Gradients (HoG). Then it blends the predictions from the two intra modes and planar mode as the final prediction.
- A DIMD flag is signaled at CU level.



# Introduction

The simplified syntax is shown as follows:

coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {	Descriptor
...	
<b>pred_mode_flag</b>	ae(v)
...	
if ( sps_dimd_enable_flag )	
<b>cu_dimd_flag</b>	ae(v)
...	
if( sps_tmp_enable_flag && cbWidth <= TMP_MaxSize && cbHeight <= TMP_MaxSize )	
<b>intra_tmp_flag</b>	ae(v)
if ( !intra_tmp_flag && !cu_dimd_flag ){	
if( sps_mip_enabled_flag )	
<b>intra_mip_flag</b>	ae(v)
if( intra_mip_flag ) {	
<b>intra_mip_transposed_flag</b> [ x0 ][ y0 ]	ae(v)
<b>intra_mip_mode</b> [ x0 ][ y0 ]	ae(v)
} else {	
if ( sps_timd_enable_flag && (sh_slice_type != I    (sh_slice_type == I && cbWidth * cbHeight <=1024 )))	
<b>intra_timd_flag</b>	ae(v)
if( sps_mrl_enabled_flag && (( y0 % CtbSizeY ) > 0 ))	
<b>intra_luma_ref_idx</b>	ae(v)
...	
}	
}	
}	

In the current design, TMP will be signaled no matter if DIMD is executed at the current CU.

# Proposed

The simplified syntax is shown as follows:

coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {	Descriptor
...	
<b>pred_mode_flag</b>	ae(v)
...	
if ( sps_dimd_enable_flag )	
<b>cu_dimd_flag</b>	ae(v)
...	
if( sps_tmp_enable_flag && cbWidth <= TMP_MaxSize && cbHeight <= TMP_MaxSize <b>&amp;&amp; !cu_dimd_flag</b> )	
<b>intra_tmp_flag</b>	ae(v)
if ( !intra_tmp_flag && !cu_dimd_flag ){	
if( sps_mip_enabled_flag )	
<b>intra_mip_flag</b>	ae(v)
if( intra_mip_flag ) {	
<b>intra_mip_transposed_flag</b> [ x0 ][ y0 ]	ae(v)
<b>intra_mip_mode</b> [ x0 ][ y0 ]	ae(v)
} else {	
if ( sps_timd_enable_flag && (sh_slice_type != I    (sh_slice_type == I && cbWidth * cbHeight <=1024 )))	
<b>intra_timd_flag</b>	ae(v)
if( sps_mrl_enabled_flag && (( y0 % CtbSizeY ) > 0 ))	
<b>intra_luma_ref_idx</b>	ae(v)
...	
}	
}	
}	

It is proposed to skip signaling the TMP flag if DIMD is executed at the current CU.

# Simulation results

Only class F enable TMP in ECM CTC.

	All Intra Main10				
	Over ECM2.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	0.00%	0.00%	100%	99%
Class A2	0.00%	0.00%	0.00%	99%	100%
Class B	0.00%	0.00%	0.00%	99%	101%
Class C	0.00%	0.00%	0.00%	99%	100%
Class E	0.00%	0.00%	0.00%	100%	100%
Overall	0.00%	0.00%	0.00%	99%	100%
Class D	0.00%	0.00%	0.00%	100%	100%
Class F	-0.05%	-0.17%	-0.08%	100%	100%

	Random Access Main 10				
	Over ECM2.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	0.00%	0.00%	100%	100%
Class A2	0.00%	0.00%	0.00%	101%	100%
Class B	0.00%	0.00%	0.00%	100%	101%
Class C	0.00%	0.00%	0.00%	100%	99%
Class E					
Overall	0.00%	0.00%	0.00%	100%	100%
Class D	0.00%	0.00%	0.00%	100%	101%
Class F	-0.03%	0.02%	-0.37%	100%	100%

	Low delay B Main10				
	Over ECM2.0				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	0.00%	0.00%	0.00%	98%	100%
Class C	0.00%	0.00%	0.00%	98%	100%
Class E	0.00%	0.00%	0.00%	102%	99%
Overall	0.00%	0.00%	0.00%	99%	100%
Class D	0.00%	0.00%	0.00%	99%	97%
Class F	-0.18%	-0.36%	-0.81%	100%	100%

# Conclusion

- Harmonization of CU-level flags of DIMD and TMP is proposed.
- Compared with ECM-2.0, it can provide  
Coding gain for AI class F: -0.05% Y, -0.17% U, -0.08% V;  
Coding gain for RA class F: -0.03% Y, 0.02% U, -0.37% V;  
Coding gain for LDB class F: -0.18% Y, -0.36% U, -0.81% V,
- No running time change in all the tests.
- It is suggested to adopt to the next version of ECM.

Thanks Alibaba for cross-checking.

Thank you

oppo