

coding_unit(x0, y0, log2CbSize) {	注释
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	
 if(slice_type != I)	(1-1)
cu_skip_flag[x0][y0]	
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
 if(intra_block_copy_enabled_flag)	(1-2)
 intra_block_copy_flag[x0][y0]	(1-3)
...	

prediction_unit(x0, y0, nPbW, nPbH) {	注释
if(cu_skip_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	
} else {	
merge_flag[x0][y0]	(2-1)
if(merge_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	
} else { /* non merge IntraBC mode syntax*/	(2-2)
intra_bc_flag	(2-3)
if(intra_bc_flag) {	(2-4)
bvd_coding(x0, y0, 2)	
bvp_flag[x0][y0]	
} else { /* non Merge Inter mode syntax*/	(2-5)
...	(2-6)
}	
}	

coding_unit(x0, y0, log2CbSize) {	注释
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	
 if(slice_type != I)	(3-1)
cu_skip_flag[x0][y0]	
nCbs = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbs, nCbs)	
else {	
if(intra_block_copy_enabled_flag)	(3-2)
intra_bc_flag[x0][y0]	(3-3)
...	

prediction_unit(x0, y0, nPbW, nPbH) {	注释
if(cu_skip_flag[x0][y0]) {	(4-1)
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	
} else if(intra_bc_flag) { /* MODE_INTRABC */	(4-2)
merge_flag[x0][y0]	(4-3)
if(merge_flag[x0][y0]) {	(4-4)
if(MaxNumMergeCand > 1)	(4-5)
merge_idx[x0][y0]	(4-6)
} else {	(4-7)
bvd_coding(x0, y0, 2)	(4-8)
bvp_flag[x0][y0]	(4-9)
}	
} else { /* MODE_INTER */	(4-10)
merge_flag[x0][y0]	
if(merge_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	
} else {	
...	

if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	
if(slice_type != I)	(3-1)
cu_skip_flag[x0][y0]	
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(intra_block_copy_enabled_flag)	(3-2)
intra_bc_flag[x0][y0]	(3-3)
...	

<code>prediction_unit(x0, y0, nPbW, nPbH) {</code>	注释
<code> if(cu_skip_flag[x0][y0]) {</code>	(4-1)
<code> if(MaxNumMergeCand > 1)</code>	
<code> merge_idx[x0][y0]</code>	
<code> } else if(intra_bc_flag) { /* MODE_INTRABC */</code>	(4-2)
<code> if(slice_type == I_SLICE)</code>	(6-1)
<code> merge_flag[x0][y0]</code>	(4-3)
<code> if(merge_flag[x0][y0] && slice_type == I_SLICE)</code>	(4-4)
<code> {</code>	
<code> if(MaxNumMergeCand > 1)</code>	(4-5)
<code> merge_idx[x0][y0]</code>	(4-6)
<code> } else {</code>	(4-7)

bvd_coding(x0, y0, 2)

(4-8)

bvp_flag[x0][y0]

(4-9)

}

}else{/* MODE_INTER */

(4-10)

merge_flag[x0][y0]

if(merge_flag[x0][y0]) {

if(MaxNumMergeCand > 1)

merge_idx[x0][y0]

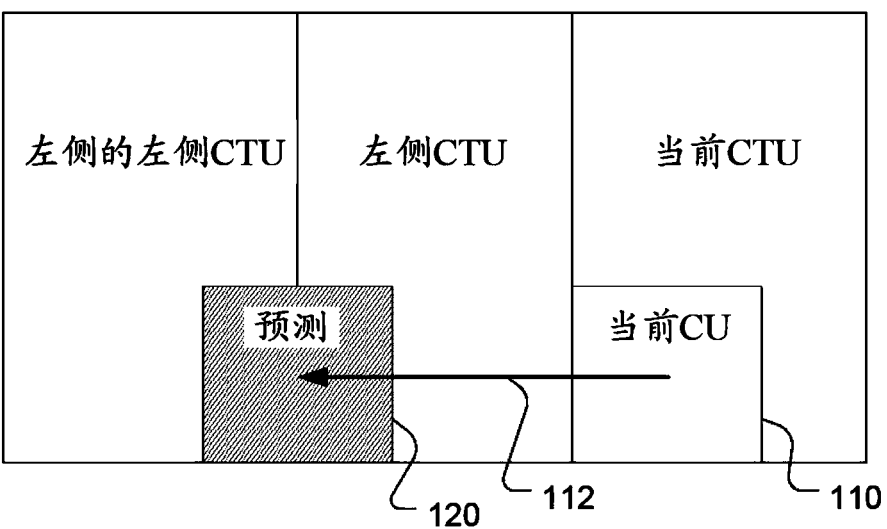
} else {

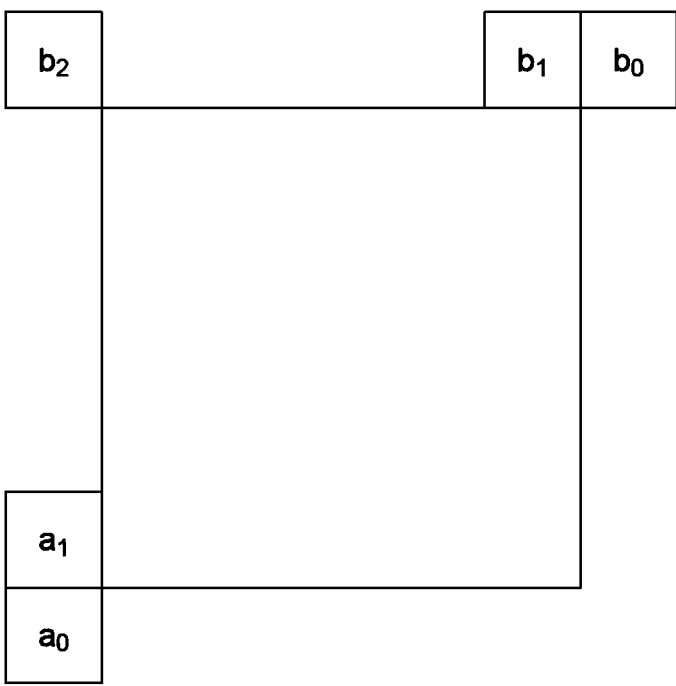
...

对照: SCM2.0	AI	RA	LB
RGB, 运动的文本及图形, 1080p	-4.5%	-5.3%	-5.5%
RGB, 运动的文本及图形, 720p	-2.4%	-2.1%	-1.6%
RGB, 混合内容, 1440p	-2.3%	-1.9%	-1.6%
RGB, 混合内容 t, 1080p	-2.0%	-2.0%	-1.9%
RGB, 动画, 720p	-0.1%	0.0%	-0.1%
RGB, 相机捕获 1080p	0.1%	0.0%	0.0%
YUV, 运动的文本及图形, 1080p	-2.4%	-4.8%	-5.3%
YUV, 运动的文本及图形 n, 720p	-1.7%	-1.6%	-1.6%
YUV, 混合内容, 1440p	-1.7%	-1.6%	-1.5%
YUV, 混合内容, 1080p	-1.6%	-1.7%	-1.6%
YUV, 动画, 720p	-0.1%	0.0%	0.0%
YUV, 相机捕获, 1080p	0.1%	0.0%	0.0%
编码时间[%]	120%	103%	102%
解码时间[%]	98%	97%	98%

对照: SCM2.0	AI	RA	LB
RGB, 运动的文本及图形, 1080p	-5.3%	-6.7%	-7.0%
RGB, 运动的文本及图形, 720p	-3.0%	-2.5%	-2.0%
RGB, 混合内容, 1440p	-2.9%	-2.4%	-2.2%
RGB, 混合内容, 1080p	-2.6%	-2.4%	-2.1%
RGB, 动画, 720p	-0.2%	0.2%	0.1%
RGB, 相机捕获, 1080p	0.0%	0.2%	0.1%
YUV, 运动的文本及图形, 1080p	-3.4%	-6.3%	-6.9%

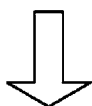
YUV, 运动的文本及图形,720p	-2.7%	-2.3%	-2.1%
YUV, 混合内容, 1440p	-2.3%	-2.1%	-2.2%
YUV, 混合内容, 1080p	-2.3%	-2.3%	-2.6%
YUV, 动画, 720p	-0.3%	0.0%	0.3%
YUV, 相机捕获, 1080p	0.0%	0.1%	0.1%
编码时间[%]	119%	101%	103%
解码时间[%]	95%	95%	99%





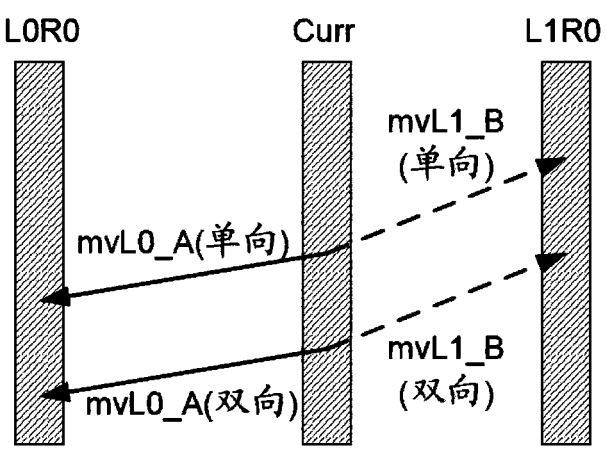
Merge_idx	L0	L1
0	mvL0_A,ref0	-
1	-	mvL1_B,ref0
2		
3		
4		

310



Merge_idx	L0	L1
0	mvL0_A,ref0	
1		mvL1_B,ref0
2	mvL0_A,ref0	mvL1_B,ref0
3		
4		

320



Merge_idx	L0	L1
0	mvL0_A,ref0	-
1	-	mvL1_B,ref0
2	mvL0_A,ref0	mvL1_B,ref0
3	-	-
4	-	-

410



Merge_idx	L0	L1
0	mvL0_A,ref0	
1		mvL1_B,ref0
2	mvL0_A,ref0	mvL1_B,ref0
3	(0,0), ref0	(0,0), ref0
4	(0,0), ref1	(0,0), ref1

420

Amvp_idx	L0	Amvp_idx	L1
0	mvL0_A	0	mvL1_A
1	mvL0_B	1	-
2	-	2	-

430



Amvp_idx	L0	Amvp_idx	L1
0	mvL0_A	0	mvL1_A
1	mvL0_B	1	(0,0)
2	(0,0)	2	-

440

开始

接收与当前图片中当前块相关联的输入数据

510

导出包括一个或多个IntraBC合并候选以及一个或多个帧间合并候选的合并候选列表，其中，所述一个或多个帧间合并候选中的每个的候选运动信息识别一个参考列表中的一个时间参考图片的一个时间参考块。

520

当前合并候选是否为
IntraBC 合并候选?

530

否

是

通过将所选择的IntraBC合并候选的候选运动信息共享为当前块的当前运动信息来编码或解码当前块的当前运动信息

540

使用由根据作为IntraBC预测子的所选择的IntraBC合并候选的候选运动信息定位的当前图片中的已编码块来编码或解码当前块

550

结束

开始

接收与当前图片中当前块相关联的输入数据

510

导出包括一个或多个IntraBC合并候选以及一个或多个帧间合并候选的合并候选列表，其中，所述一个或多个帧间合并候选中的每个的候选运动信息识别一个参考列表中的一个时间参考图片的一个时间参考块。

520

当前合并候选是否为
IntraBC 合并候选?

530

否

是

通过将所选择的IntraBC合并候选的候选运动信息共享为当前块的当前运动信息来编码或解码当前块的当前运动信息

540

使用由根据作为IntraBC预测子的所选择的IntraBC合并候选的候选运动信息定位的当前图片中的已编码块来编码或解码当前块

550

结束