



# **Non-CE4: AMVR cost calculation modification in motion estimation stage**

**Yucheng Sun**  
**2019/3/17**

- Proposed Modification
- Experimental Results
- Conclusion

- In VTM 4.0
  - AMVR cost calculation omit mvp\_flag cost

```
int iMvBits = m_auIMVPIIdxCost[iMVPIIdx][AMVP_MAX_NUM_CANDS];
```

...

```
iMvBits += m_pcRdCost->getBitsOfVectorWithPredictor( cTestMv[iMVPIIdx].getHor(),  
cTestMv[iMVPIIdx].getVer(), cStruct.imvShift );
```

```
uiDist += m_pcRdCost->getCostOfVectorWithPredictor( cTestMv[iMVPIIdx].getHor(),  
cTestMv[iMVPIIdx].getVer(), cStruct.imvShift );
```

**Only calculate SATD and MVD cost**

- In VTM 4.0
  - AMVR cost calculation omit mvp\_flag cost

```
int iMvBits = m_auIMVPIIdxCost[iMVPIIdx][AMVP_MAX_NUM_CANDS];
```

...

```
iMvBits += m_pcRdCost->getBitsOfVectorWithPredictor( cTestMv[iMVPIIdx].getHor(),  
cTestMv[iMVPIIdx].getVer(), cStruct.imvShift );
```

```
uiDist += m_pcRdCost->getCost(iMvBits);
```

**iMvBits calculate MVD + mvp\_flag**

- AMVR bits calculation double calculate MVD bits

```
ruiBits += iBestBits;
```

**calculate MVD bits**

```
ruiCost = uiBestDist - m_pcRdCost->getCost(iBestBits) + m_pcRdCost->getCost(ruiBits);
```

```
ruiBits += m_pcRdCost->getBitsOfVectorWithPredictor(rcMv.getHor(), rcMv.getVer()  
cStruct.imvShift);
```

**Double calculate MVD bits**

- AMVR bits calculation double calculate MVD bits

```
ruBits += iBestBits;
```

**calculate MVD bits**

```
ruCost = uiBestDist - m_pcRdCost->getCost(iBestBits) + m_pcRdCost->getCost(ruBits);
```

**Delete**

	Random access Main10				
	Over VTM-4.0				
	Y	U	V	EncT	DecT
Class A1	-0.03%	-0.09%	0.00%	100%	100%
Class A2	-0.03%	-0.06%	-0.01%	100%	99%
Class B	-0.01%	-0.13%	-0.09%	100%	100%
Class C	-0.03%	-0.08%	-0.16%	100%	102%
Class E					
<b>Overall</b>	-0.02%	-0.10%	-0.07%	100%	100%
Class D	-0.01%	-0.01%	-0.10%	101%	104%
Class F	-0.01%	-0.03%	-0.03%	100%	103%

	Low delay B Main10				
	Over VTM-4.0				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	-0.02%	-0.06%	-0.21%	100%	101%
Class C	0.03%	0.06%	-0.21%	100%	101%
Class E	-0.01%	0.01%	-0.01%	100%	101%
<b>Overall</b>	-0.00%	-0.00%	-0.16%	100%	101%
Class D	0.00%	0.09%	-0.28%	100%	101%
Class F	-0.09%	-0.05%	-0.05%	101%	101%

	Low delay P Main10				
	Over VTM-4.0				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	0.02%	-0.25%	-0.19%	100%	104%
Class C	-0.01%	-0.14%	-0.37%	99%	98%
Class E	0.01%	0.06%	0.18%	99%	100%
<b>Overall</b>	0.01%	-0.14%	-0.16%	100%	101%
Class D	0.01%	-0.13%	0.00%	96%	99%
Class F	0.03%	0.04%	0.09%	98%	101%

Thank HHI for the cross-checking

- AMVR cost calculation **double calculate MVD bits** and **omit mvp\_flag cost**
  - Influence the **MV decision** in motion estimation stage.
  - Influence the **cost calculation in other tools** when invoke the AMVR bits result. e.g. AMVR bi-prediction
- Experimental results reportedly show
  - -0.02%, -0.10%, -0.07% compared to VTM-RA
  - -0.00%, -0.00%, -0.16% compared to VTM-LDB
  - 0.01%, -0.14%, -0.16% compared to VTM-LDP
- Based on the above results, it is suggested to modify AMVR cost calculation in VTM platform



# THANKS !

公司总部：杭州市滨江区东流路700号

股票代码：002415

客服电话：400-700-5998

[www.hikvision.com](http://www.hikvision.com)

© 杭州海康威视数字技术股份有限公司

分公司：北京、长沙、成都、重庆、长春、福州、广州、贵阳、杭州、合肥、哈尔滨、呼和浩特、济南、昆明、兰州、南昌、  
南京、南宁、宁波、青岛、上海、沈阳、深圳、石家庄、太原、天津、武汉、厦门、乌鲁木齐、西安、郑州.....

海外分公司：洛杉矶、香港、阿姆斯特丹、孟买、圣彼得堡、迪拜、新加坡.....