

LCEC RDOQ speedup

Vadim Seregin, Jianle Chen

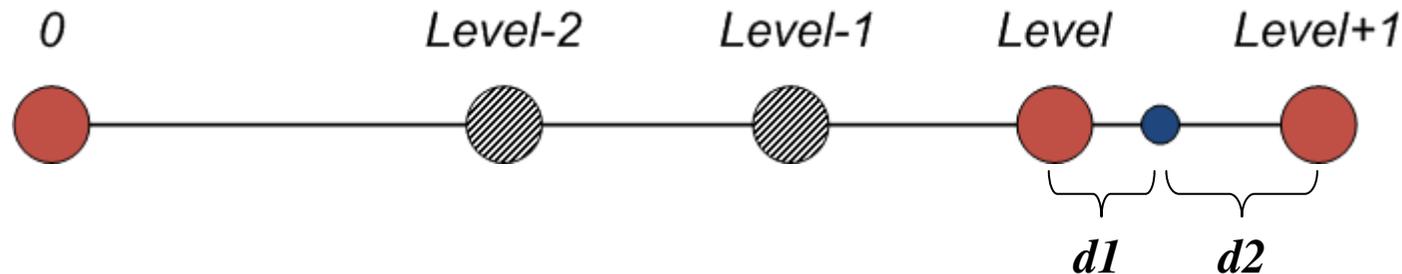
Samsung Electronics

Jan. 23th, 2011

Introduction

❖ Current RDOQ implementation

- Arithmetic entropy coding condition
 - Up to three quantized candidates for each coefficient
- LCEC entropy coding condition
 - Up to five quantised candidates for each coefficient



❖ Problem

- There is unnecessary coefficient checking in LCEC RDOQ

❖ Solution

- Remove two unnecessary quantised candidates in LCEC RDOQ harmonising implementation with HE RDOQ

Implementation details

- ❖ Only one new line is added

```
#if LCEC_RDOQ_SPEEDUP
```

```
    Bool bLowerInt = ( ( lLevelDouble - ((Int64)uiLevel << iQBits ) ) < ( (Int64)1 << ( iQBits  
        - 1 ) ) ); // additional line for checking whether d1 is smaller than d2
```

```
    uiMaxLevel = bLowerInt ? uiLevel : uiLevel + 1;
```

```
    uiMinLevel = Max(1, uiLevel);
```

```
#else
```

```
    uiMaxLevel = uiLevel + 1;
```

```
    uiMinLevel = Max(1,(Int)uiLevel - 2);
```

```
#endif
```

Experimental results and conclusion

- ❖ Average BD-rate saving against HM 0.9 anchor

		RDOQ speedup				
		Y	U	V	EncT	DecT
Intra	LoCo	0.0	0.0	0.0	81%	100%
RA	LoCo	0.1	0.1	0.1	93%	103%
LD	LoCo	0.0	0.1	0.0	92%	96%

- ❖ Proposed speedup of RDOQ harmonises implementation with HE RDOQ
- ❖ Complexity reduction up to 19% with negligible performance loss
- ❖ Suggestion
 - Use proposed RDOQ speedup in HM