TOSHIBA

Leading Innovation >>>

JCTVC-D152
Adaptive scaling for bit depth compression on IBDI

Takeshi Chujoh
Tomoo Yamakage

TOSHIBA CORPORATION

January 22, 2011

Summary

- **Adhoc Group on memory compression**

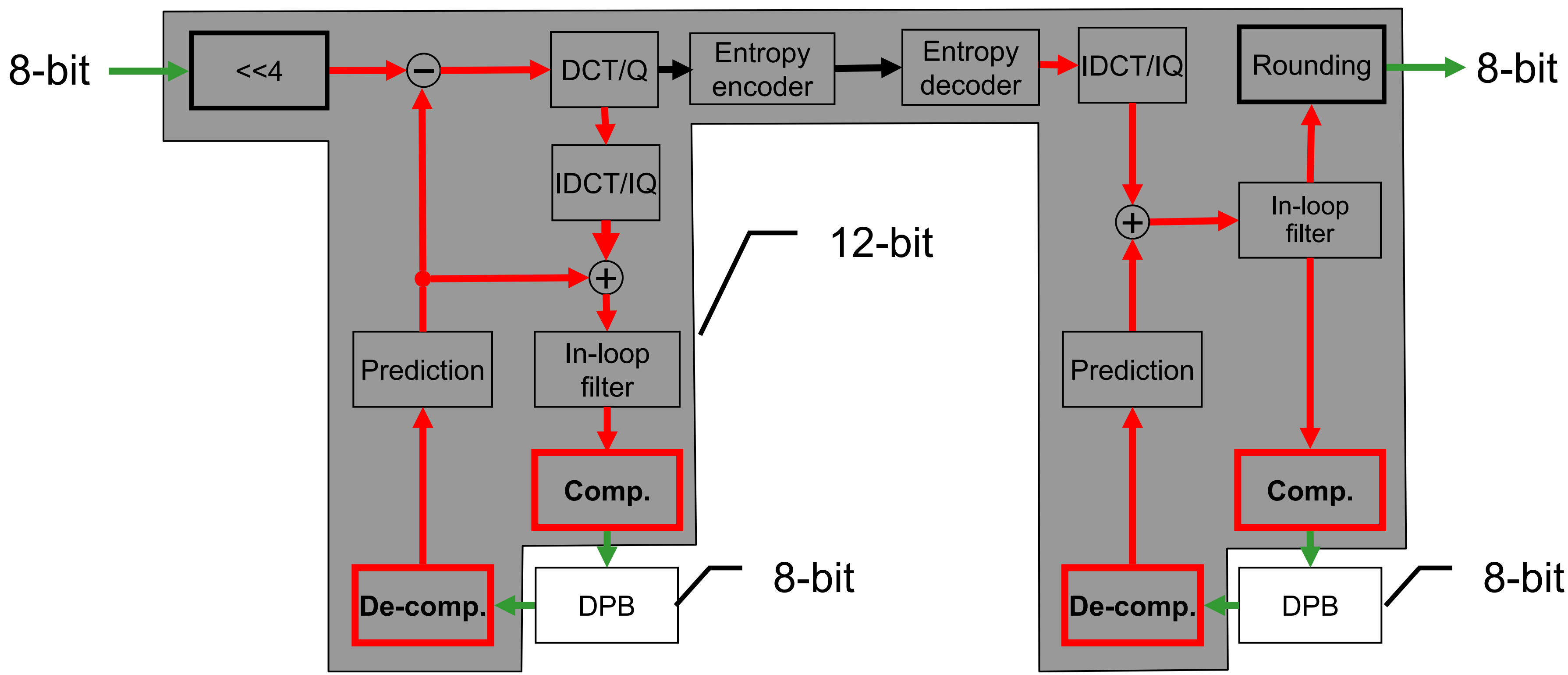
- **Adaptive scaling for bit depth compression on IBDI**
  - Fixed length format by 4x4 block
  - Definition of compression distortion

- **Solution of fixed rounding problem**

- **Experimental results**
  - Loss is 0.71% compared to IBDI
  - Overhead of complexity is negligible.

Bit depth compression on IBDI

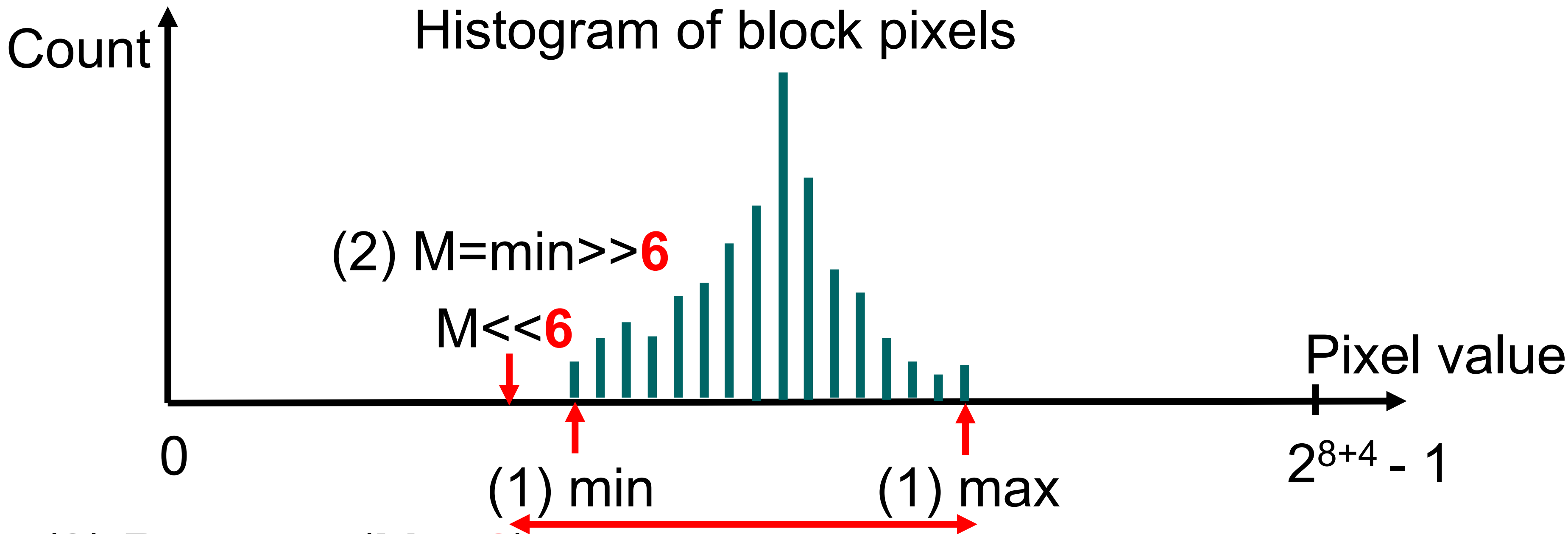
- **An example of internal 12-bit**



Adaptive scaling

- **Dynamic range adaptive scaling**
- **Scaling process by 4x4 block**
- **Loss less compression on 8-bit depth level**
  - Results surely better than fixed scaling
- **Complexity is negligible**
- **Fixed length format**
- **Definition of compression destortion**

Encoding process (4x4 block)




```
(3) R=max – (M<<6);
(4) for (S=0; R>=(128<<S) && S<4; S++);
(5) if (S==4) {
        P[0] = (pixel_value[0]<8) 1: (pixel_value[0]+8)>>4;
        for(i=1;i<16;i++) P[i] = (pixel_value[i]+8)>>4;
    } else
        for(i=0;i<16;i++) P[i] = (pixel_value[i] – (M<<6))>>S;
```

Fixed length format (128-bit)

adaptive_scaling_block( ) {	
P[0]	u(8)
if (P[0]!=0) {	
for (i=1;i<16;i++) {	
P[i]	u(8)
}	
} else {	
S	u(2)
M	u(6)
for (i=0; i<16; i++) {	
P[i]	u(7)
}	
}	
}	

128 bits,
If Q=4

(8+120) bits,
If Q<4

Decoding process

```
if (P[0] != 0) {
        for (i=0;i<16;i++)
                D[i] = P[i] <<4;
} else {
   for (i=0;i<16;i++)
        D[i] = (P[i] << S) + (M<<6) + ((S!=0)? (1<<(S-1)):0);
}
```

Definition of compression distortion

(1) Find min and max in 4x4 block

(2) Decide the scaling value S from difference between max and min.

```
for (Q=0; max – (min & ~0x3f)>=(128<<Q) && Q<4; Q++);
```

(3) Control bit depth of pixels according to S.

```
if (S==4) {
        D[0] = (pixel_value[0]<8)? 16: (pixel_value[0]+8) & ~0xf;
        for (i=1;i<16;i++) D[i] = (pixel_value[i]+8) & ~0xf;
} else if (S==0) {
        for (i=0;i<16;i++) D[i] = pixel_value[i];
} else {
        for (i=0;i<16;i++) D[i] = (pixel_value[i] & (~0<<S)) + (1<<(S-1));
}
```

Fixed rounding problem

- **IBDI with fixed rounding of frame memory is worse than IBDI off.**

- **TMuC software**

$$D = \sum_i ((org[i] << 4) - ref[i]) \times (org[i] << 4) - ref[i])) >> 4 \qquad (1)$$

- **KTA software**

$$D = \sum_i (org[i] - (ref[i] + 8) >> 4) \times (org[i] - (ref[i] + 8) >> 4) \qquad (2)$$

Experimental results

	IBDI with Eq. 2		IBDI with fixed rounding and Eq.1		IBDI with fixed rounding and Eq. 2		IBDI off	
	Random access	Low delay	Random access	Low delay	Random access	Low delay	Random access	Low delay
Class A	0.1	N/A	1.3	N/A	0.9	N/A	1.5	N/A
Class B	0.2	0.4	1.6	4.1	1.1	2.2	2.1	3.5
Class C	0.2	0.2	1.0	2.5	0.8	1.5	1.3	2.4
Class D	0.1	0.2	0.6	1.6	0.5	1.1	1.0	1.6
Class E	N/A	0.4	N/A	27.1	N/A	8.5	N/A	11.4
Total	**0.2**	**0.3**	**1.2**	**7.4**	**0.8**	**2.9**	**1.5**	**4.2**

* All conditions include TPE on

Test condition

- **Configurations are base on JCTVC-C500:**
  - Implementation on TMuC 0.9-ahg-memory branch
  - 12-bit to 8-bit (Anchor including IBDI)
    - High coding efficiency, random access configuration
    - High coding efficiency, low delay configuration
  - Common test condition is specified in JCTVC-C500

- **Evaluation criteria**
  - Measure impact on bitrate/PSNR using provided data. Use 4-point BD-PSNR and BD-Rate.
  - Memory compression ratio.
  - Memory access measures
  - Complexity (encoding and decoding times)

Experimental results (Summary)

	Random access			Low delay		
	Adaptive scaling	Fixed rounding	IBDI off	Adaptive scaling	Fixed rounding	IBDI off
Class A	0.25	0.93	1.47	N/A	N/A	N/A
Class B	0.39	1.12	2.13	0.86	2.23	3.54
Class C	0.28	0.76	1.28	0.53	1.51	2.40
Class D	0.23	0.52	0.96	0.36	1.08	1.61
Class E	N/A	N/A	N/A	3.33	8.48	11.38
Total	**0.30**	**0.84**	**1.50**	**1.22**	**2.93**	**4.25**

Adaptive(0.71) < Fixed(1.89) < IBDI off(2.87)

* All conditions include TPE on

Complexity

- **Encoding and decoding time**
  - Increase rate (%) of average encoding and decoding times on IBDI

Proponent	Adaptive scaling with fixed length format		Adaptive scaling by distortion definition		Fixed rounding	
Condition	Random access	Low delay	Random access	Low delay	Random access	Low delay
Encoding Time	0.46	0.41	0.34	0.18	1.11	1.19
Decoding Time	7.98	7.88	7.10	4.07	8.11	8.24

Not significantly increase

Memory access

- **Average motion compensation memory bandwidth saving rate (%)**
  - Both Luma and Chroma block size define 4x4
  - Rate defines 0.667

	8bit/ 8bit	32bit/ 64bit	32bit/ 128bit	64bit/ 128bit	64bit/ 256bit	64bit/ 512bit	64bit/ 256bit FIFO	64bit/ 512bit FIFO
RA	-11.1	-21.1	-25.1	-30.3	-39.2	-48.9	-36.8	-37.3
LD	-9.8	-19.9	-23.7	-29.5	-39.4	-49.9	-39.5	-40.7

Conclusion

- **Adaptive scaling for bit depth compression on IBDI**
  - That loss is always smaller than loss of fixed rounding
  - Increase of complexity is negligible
- **Solution of fixed rounding problem**
- **Cross-checked by JCTVC-D281 (NEC)**
- **This is one of the solution of bit-depth compression of DPB on IBDI**

TOSHIBA

Leading Innovation >>>