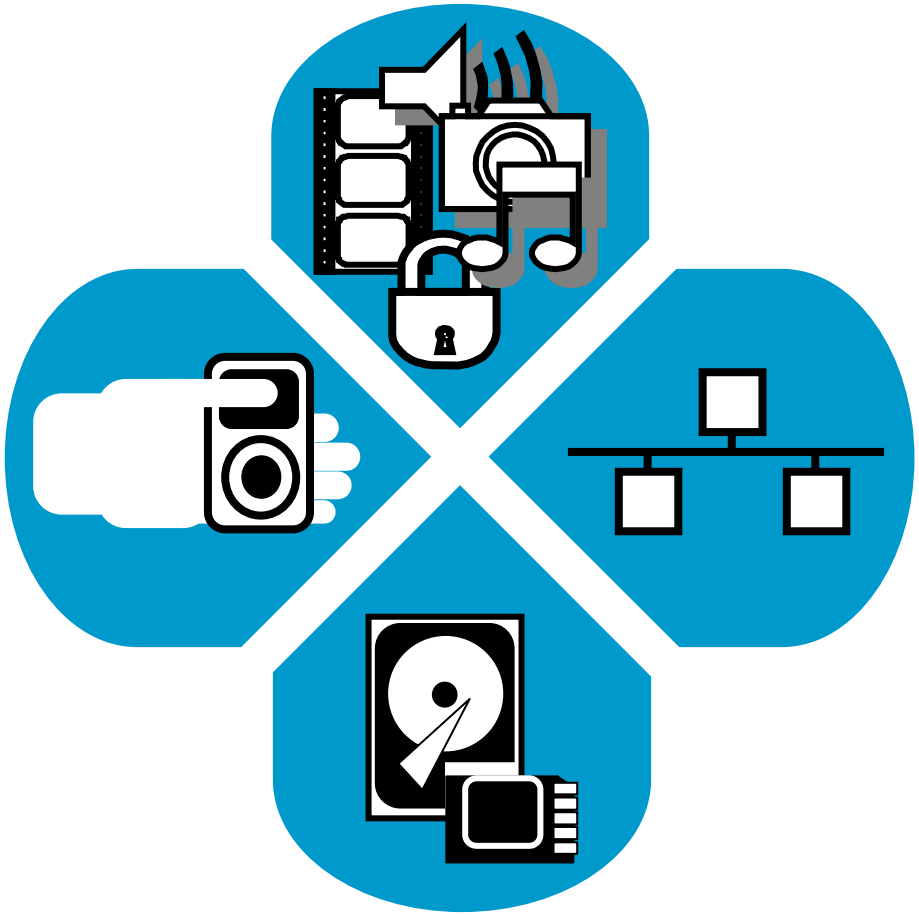


zenverge

convergence made simple™

Unified Scaling for 10-bit to 8-bit Reference Frame Compression

Dzung Hoang

JCTVC-E432

March 2011

Summary

- We propose a reference frame compression (RFC) technique to:
  - code and decode 10-bit video using same frame buffer memory as 8-bit video, or
  - code and decode 8-bit video using IBDI (10 bits internal) using same frame buffer memory as 10-bit video
- The proposal is a (non-trivial) modification of JCTVC-D035, which itself is a (non-trivial) modification of Toshiba's JCTVC-C075.
- We compare to HM 2.0 anchor with Fixed Rounding (JCTVC-D045,-D152) and HM 2.0 w/o IBDI.
- The Y BD-rate losses compared to HM 2.0 on LD-HE and RA-HE configurations are 1.4% and 0.3%, resp. Chroma BD-rate performance is within 94% of gain of IBDI.
- If PSNR is computed before RFC, the losses are 1.2% and 0.1%.
- Encoder run time is 102% of HM 2.0 and decoder run time is 113%.
- Cross-checked by Santa Clara University in JCTVC-E463.

Unified Scaling with Adaptive Offset

zenverge
convergence made simple™

- **Introduced in JCTVC-D035**
  - Based upon Toshiba JCTVC-C075
  - Remove fixed scaling
  - Encoder computes scaling factor S
  - Encoder computes reconstruction offset as the average of quantizer error and transmits offset using S bits
  - Encoder computes and transmits the block minimum value (spatial predictor) using 12-S bits

Zenverge RFC Format

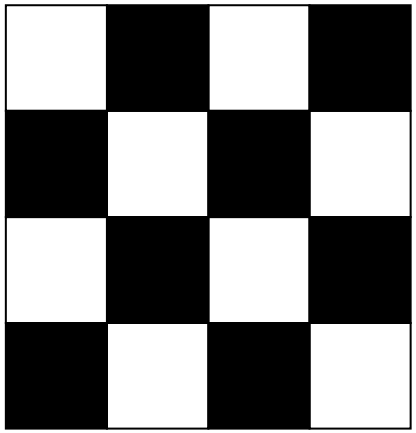
- Store quantizer scale S using 2 bits.
- Store minimum and offset using total of 10 bits.
- Indicate position of minimum sample using 4 bits.
- Store 7 pixels using 8 bits (P8 set) and 8 pixels using 7 bits (P7 set).

u(2);	/* S: [0..3] */
u(10-S);	/* M */
u(S);	/* offset */
u(4);	/* M_index */
for (i=0; i<8; i++) u(7);	/* P7 array */
for (i=0; i<7; i++) u(8);	/* P8 array */

Checkerboard Sampling

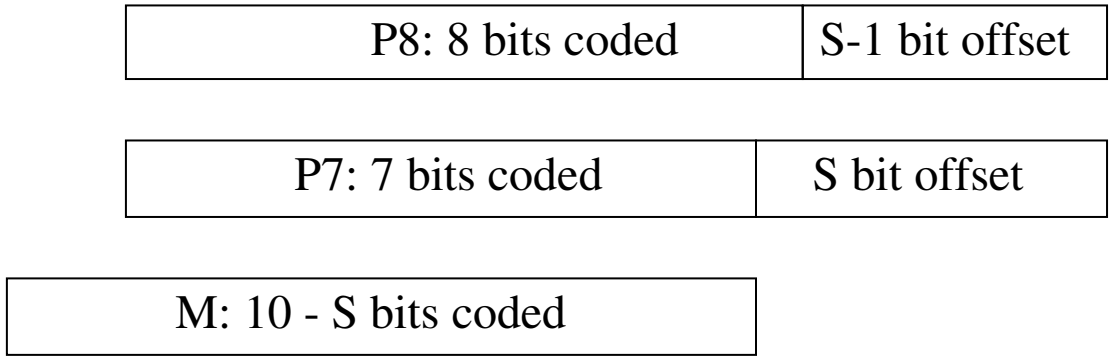
zenverge
convergence made simple™

- Position of 8-bit and 7-bit residuals are arranged in checkerboard pattern.
- 8-bit residuals have the same color as the minimum sample.
- Avoids potential artifacts due to “clumping.”



Offsets for P8 and P7

- **Conceptually apply reconstruction offset as fraction added to quantized residual before inverse quantization.**



Zenverge RFC Decompression


```
S' = max(0, S-1); offset' = offset >> 1
min_color = checkerboard color of M_index
i7 = i8 = 0
for (i=0; i<16; i++) {
    if (i == M_index)
        D[i] = M + offset
    else if (min_color == checkerboard color of position i)
        D[i] = (P8[i8++] << S') + M + offset'
    else
        D[i] = (P7[i8++] << S) + M + offset
}
```


Computation of Offset

- Instead of exhaustive search for best offset, we derive an analytical formula based upon least squares method.
- offset = floor(3/64*(E_8+E_7+E_M))
  - E_8 = sum squared error of 8-bit residuals
  - E_7 = sum squared error of 7-bit residuals
  - E_M = sum squared error of minimum

Zenverge RFC Compressor

```
M = (minimum pixel value in block)
M_index = index of first occurrence of M in block; min_color = color of M_index
R = (maximum pixel value in block)
for (S=0; (R>>S)-(M>>S)>=128; S++);
S′ = max(0, S-1); mask = (1 << S) − 1; mask′ = (1 << S′) − 1
M = M & ~mask; err = i7 = i8 = 0
for (i=0; i<16; i++) {
    if ( i == M_index )
        err += 2*(pixel_value[i] & mask)
    else if ( min_color == color for raster scan position i )
        P8[i8++] = (pixel_value[i] - M) >> S′; err += (pixel_value[i] & mask′)
    else
        P7[i8++] = (pixel_value[i] - M) >> S; err += 2*(pixel_value[i] & mask)
}
offset = 3*err/64
```


Implementation Details

- Added RFC on top of HM-2.0-ahg-memory branch.
  - decoder incurs extra run time from memory bandwidth computations
- Added MD5 sum computation at run time to avoid dumping YUV files.
  - has negligible effect on encoder run times
  - has significant effect on decoder run times
- For decoder, run times recomputed after disabling MD5 sum.
- Compared against Fixed Rounding using HM-2.0-dev-toshiba.
  - does not compute memory bandwidth
  - run times are lower
- In encoder, PSNR was computed after RFC compression.
- Decoder was later modified to dump frames before RFC to compute PSNR before RFC compression.

Results for LD-HE

- **Compared to HM 2.0 anchor**

	Zenverge RFC			Fixed Rounding			HM 2.0 w/o IBDI		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
Class B	0.7	0.0	0.2	2.5	7.1	9.2	3.2	10.9	12.5
Class C	0.6	0.5	0.4	1.4	3.8	3.7	1.8	5.0	5.6
Class D	0.3	0.3	0.6	0.9	7.5	7.7	1.1	9.1	9.6
Class E	5.0	0.4	0.6	8.8	20.8	23.1	7.2	18.2	14.8
All	**1.4**	**0.3**	**0.4**	**3.0**	**9.0**	**10.1**	**3.1**	**10.4**	**10.5**
Enc Time	102%			102%			100%		
Dec Time	113%			77%			99%		

Results for LD-HE

- **Compared to HM 2.0 w/o IBDI (8-bit)**

	HM 2.0 with IBDI			Zenverge RFC			Fixed Rounding		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
Class B	-3.1	-9.4	-10.5	-2.3	-9.4	-10.3	-0.6	-3.3	-2.8
Class C	-1.8	-4.7	-5.2	-1.2	-4.2	-4.8	-0.5	-1.1	-1.8
Class D	-1.1	-7.8	-8.2	-0.8	-7.6	-7.7	-0.1	-1.2	-1.3
Class E	-6.7	-15.4	-12.9	-2.0	-15.0	-12.4	1.5	2.5	7.8
All	**-2.9**	**-9.0**	**-9.1**	**-1.6**	**-8.7**	**-8.6**	**-0.7**	**-1.8**	**-1.9**

Results for RA-HE

- Compared to HM 2.0 anchor
- Class A8/A10 contains 8/10 bit sequences

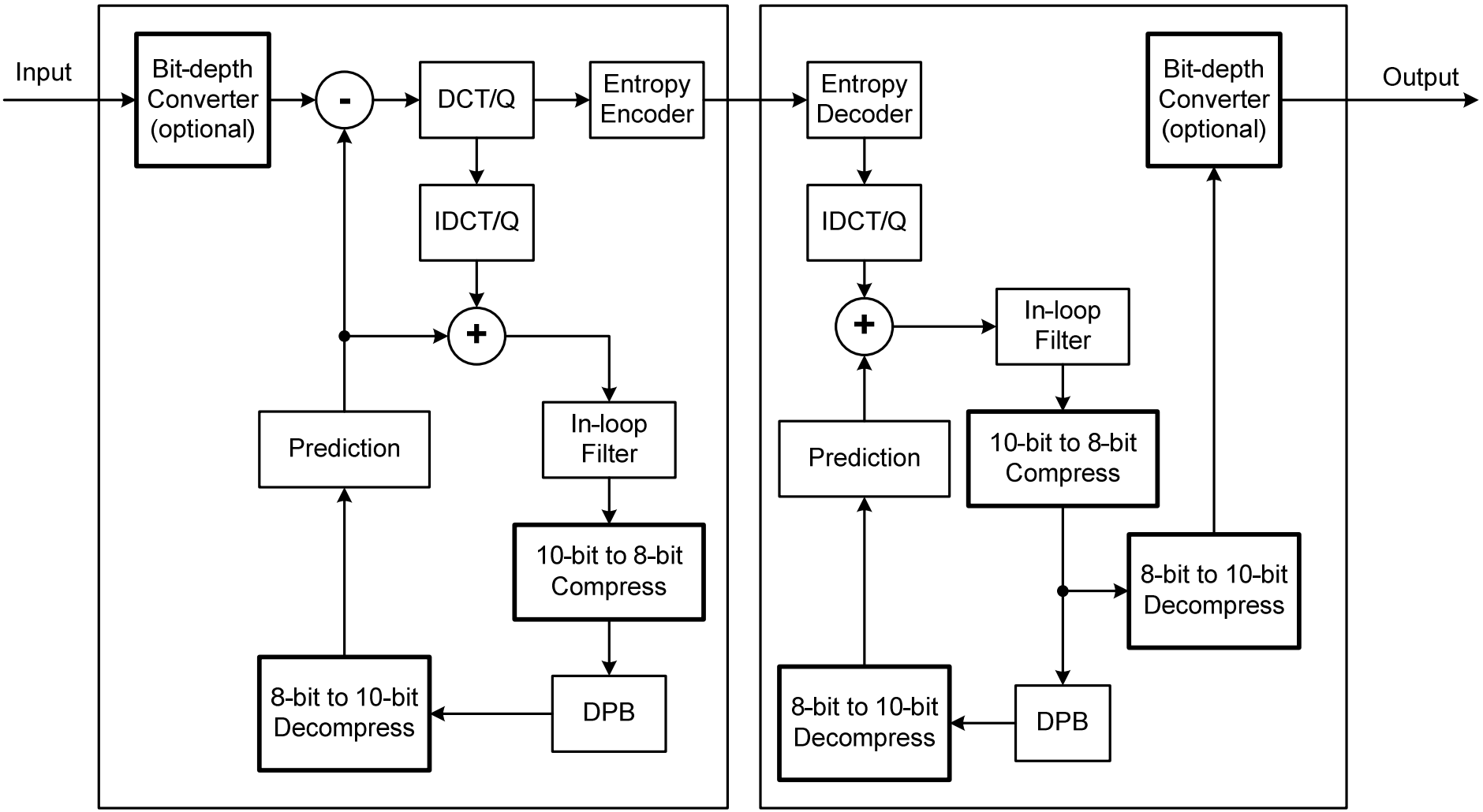
	Zenverge RFC			Fixed Rounding			HM 2.0 w/o IBDI		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
Class A8	0.2	0.0	0.0	1.0	2.7	3.3	1.5	4.3	5.2
Class A10	0.4	0.2	-0.1	1.2	13.4	21.7			
Class A	0.3	0.1	-0.1	1.1	8.0	12.5			
Class B	0.3	0.0	0.0	1.3	3.7	4.5	2.5	7.2	7.6
Class C	0.2	0.1	0.0	0.8	1.9	1.9	1.4	3.1	3.5
Class D	0.3	0.2	0.0	0.7	2.1	1.9	1.0	2.7	3.1
All	**0.3**	**0.1**	**0.0**	**1.0**	**3.9**	**5.2**	**1.7**	**4.5**	**5.0**
Enc Time	102%			102%			100%		
Dec Time	113%			77%			97%		

Results for RA-HE

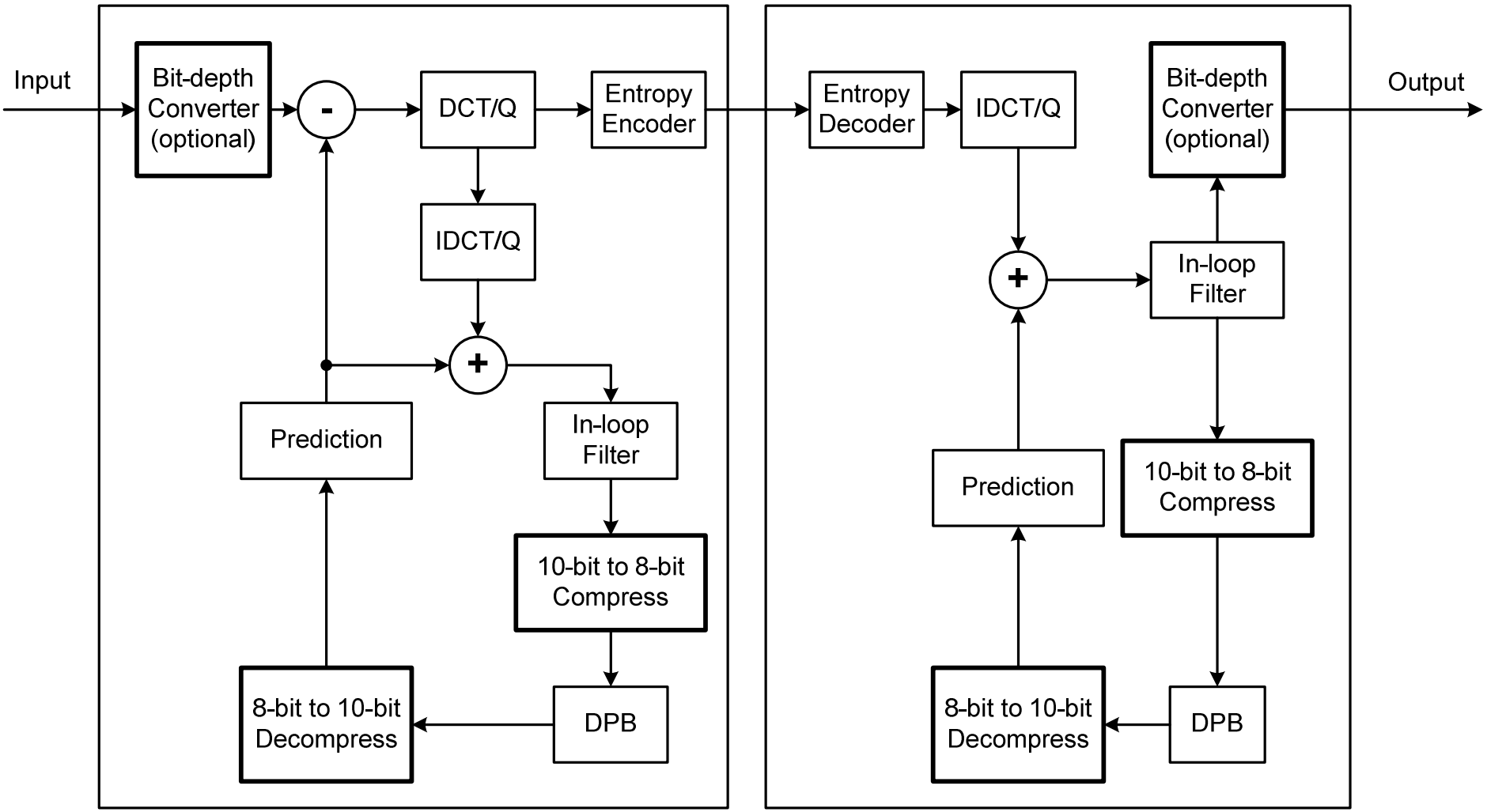
- Compared to HM 2.0 w/o IBDI (8-bit)
- Class A8 contains only 8-bit sequences

	HM 2.0 with IBDI			Zenverge RFC			Fixed Rounding		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
Class A8	-3.1	-9.4	-10.5	-2.3	-9.4	-10.3	-0.6	-3.3	-2.8
Class B	-1.8	-4.7	-5.2	-1.2	-4.2	-4.8	-0.5	-1.1	-1.8
Class C	-1.1	-7.8	-8.2	-0.8	-7.6	-7.7	-0.1	-1.2	-1.3
Class D	-6.7	-15.4	-12.9	-2.0	-15.0	-12.4	1.5	2.5	7.8
All	**-2.9**	**-9.0**	**-9.1**	**-1.6**	**-8.7**	**-8.6**	**-0.7**	**-1.8**	**-1.9**

Computing PSNR After RFC



Computing PSNR Before RFC



Results with PSNR before RFC

- **Fixed Rounding already computes PSNR before rounding**

	LD-HE			RA-HE		
	Y	Cb	Cr	Y	Cb	Cr
Class A				0.2	0.1	-0.1
Class B	0.6	0.0	0.2	0.1	0.0	0.0
Class C	0.4	0.4	0.4	0.1	0.0	0.0
Class D	0.1	0.3	0.5	0.2	0.2	0.0
Class E	4.7	0.4	0.6			
All	**1.2**	**0.3**	**0.4**	**0.1**	**0.1**	**0.0**

Memory Bandwidth LD-HE

zenverge
convergence made simple™

Per AHG agreement, memory bandwidth computed for QP=22.

	Low Delay High Efficiency (Zenverge RFC vs HM 2.0)							
	Memory bandwith increase %							
	8/8	32/ 64	32/128	64/128	64/256	64/512	64/256 FIFO	64/512 FIFO
Class B	7.9%	-7.5%	-18.7%	-23.1%	-28.6%	-46.0%	-34.2%	-38.8%
Class C	10.8%	-7.3%	-20.5%	-25.6%	-32.9%	-52.7%	-42.1%	-47.6%
Class D	20.6%	-3.0%	-18.6%	-24.6%	-33.6%	-55.3%	-43.5%	-50.0%
Class E	-8.9%	-15.9%	-23.4%	-25.6%	-35.8%	-51.1%	-35.6%	-39.1%
All	**8.6%**	**-7.9%**	**-20.0%**	**-24.6%**	**-32.3%**	**-51.0%**	**-38.8%**	**-43.9%**

Memory Bandwidth RA-HE

	Random Access High Efficiency (Zenverge RFC vs HM 2.0)							
	Memory bandwith increase %							
	8/8	32/ 64	32/128	64/128	64/256	64/512	64/256 FIFO	64/512 FIFO
Class A	5.9%	-9.5%	-21.1%	-25.2%	-33.1%	-50.9%	-32.8%	-37.5%
Class B	4.7%	-9.1%	-18.9%	-22.8%	-27.8%	-44.3%	-31.6%	-35.9%
Class C	8.3%	-7.8%	-19.9%	-24.8%	-31.7%	-50.6%	-38.0%	-43.2%
Class D	16.6%	-3.4%	-17.8%	-23.7%	-31.6%	-52.7%	-40.0%	-45.8%
All	**7.5%**	**-8.1%**	**-19.4%**	**-23.8%**	**-30.0%**	**-47.0%**	**-35.4%**	**-39.8%**

Conclusion

- The Y BD-rate losses compared to HM 2.0 on LD-HE and RA-HE configurations are 1.4% and 0.3%, resp. Chroma BD-rate performance is within 94% of gain of IBDI.
- Memory bandwidth can be reduced by 51% (LD-HE) and 47% (RA-HE) using RFC assuming 64-bit alignment and 512-bit burst. This is greater than the reduction implied by RFC (20%).
- Results indicate that Fixed Rounding performs poorly, that is not a good solution.

Recommendations

- Good potential to unify E432 and Toshiba's E133. Chujoh-san indicated willingness to do so.
- Should investigate using more than 2 bits of IBDI for 8-bit video and also 2 or more bits of IBDI for 10-bit video sequences. From D035, 12-bit IBDI with RFC achieved -3.2% Y BD-rate delta.
- We can generalize the scheme for N-bit to 8-bit, where N can vary from 9 to 12, for example. We feel such a generalized scheme would be desirable for standardization, as it would be more flexible.
- Recommend continuation of AHG and CE to study unified approach and generalization for N-to-8.