



PSNR Computation on R'G'B' Color System

Dzung Hoang JCTVC-D040



Executive Summary



- **Current HM computes PSNR for Y' , C_B , and C_R separately. Many contributions report BD-rate results only for Y' . When BD-rate results for Y' , C_B , and C_R differ, it's hard to make good conclusions. Adhoc combinations of PSNR for Y' , C_B , and C_R are suspect.**
- **We propose additional computation of PSNR based upon $R'G'B'$ color system. Then we can get a single PSNR result that combines luma and chroma components in a meaningful way.**
- **We examine a subset of results from JCTVC-D035 to illustrate the new measure.**

- **Digital video capture and display devices use standardized $R'G'B'$ color systems.**
- **Digital video compression works on $Y'C_B C_R$ color system.**
- **Quality, subjective or objective, should be performed with display technology in mind.**
- **It may be desirable to compute objective distortion in $R'G'B'$.**

How to Compute R'G'B' PSNR



1. Upsample $Y'C_B C_R$ from 4:2:0 to 4:4:4
2. Convert 4:4:4 $Y'C_B C_R$ to R'G'B'
3. Compute MSE of R', G', B'
4. Average MSE of R', G', B' to form MSE of R'G'B'
5. Compute PSNR from MSE

- Use polyphase filter coefficients borrowed from MSSG to convert 4:2:0 to 4:2:2 and then to 4:4:4
- Vertical polyphase filter
 - [3, -16, 67, 227, -32, 7]/256
 - [7, -32, 227, 67, -16, 3]/256
- Horizontal polyphase filter
 - [0, 0, 256, 0, 0]/256
 - [21, -52, 159, 159, -52, 21]/256

Convert $Y'C_B C_R$ to $R'G'B'$



- **There are different $Y'C_B C_R$ and $R'G'B'$ color systems**
 - i.e., Rec. ITU-R BT.601, Rec. ITU-R BT.709, and SMPTE 240M
- **Color space conversion matrix for Rec. ITU-R BT.709 with R' , G' , and B' in the range $[0, 1]$:**

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 0.00456621 & 0.0 & 0.00703036 \\ 0.00456621 & -0.00083627 & -0.00208984 \\ 0.00456621 & 0.00828393 & 0.0 \end{bmatrix} \bullet \left(\begin{bmatrix} Y' \\ C_B \\ C_R \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

- **Definition of PSNR for 8-bit unsigned:**

$$\text{PSNR} = 10 \log \left(\frac{255^2}{\text{MSE}} \right)$$

- **HM computes average of frame PSNR:**

$$\text{APSNR}_{\text{seq}} = \frac{\sum_i \text{PSNR}_{\text{frame}}(i)}{N}$$

Compute PSNR of R'G'B'



- **R'G'B' clipped to [0, 1]**
- **Definition of PSNR for [0, 1] data:**

$$\text{PSNR} = 10 \log \left(\frac{1}{\text{MSE}} \right)$$

IBDI versus no-IBDI for LD-HE



		Y	Cb	Cr	R'	G'	B'	R'G'B'
Class B 1080p	Kimono	-2.2	-4.3	-4.8	-3.3	-2.4	-3.8	-3.3
	ParkScene	-2.6	-7.0	-10.1	-5.2	-2.9	-6.2	-5.0
	Cactus	-4.0	-5.3	-6.7	-5.5	-4.1	-4.9	-5.0
	BasketballDrive	-3.1	-9.1	-7.2	-5.6	-3.6	-6.9	-5.7
	BQTerrace	-5.1	-25.0	-31.0	-11.3	-5.5	-14.2	-11.1
	Average	-3.4	-10.2	-11.9	-6.2	-3.7	-7.2	-6.0
Class C WVGA	BasketballDrill	-2.8	-6.4	-5.3	-4.5	-3.2	-5.3	-4.6
	BQMall	-3.6	-9.1	-9.8	-6.3	-4.0	-6.7	-6.0
	PartyScene	-1.8	-9.7	-10.7	-5.0	-2.3	-6.2	-4.8
	RaceHorses	-1.0	-2.9	-2.7	-1.9	-1.1	-2.2	-1.9
	Average	-2.3	-7.0	-7.2	-4.4	-2.6	-5.1	-4.3
Class D WQVGA	BasketballPass	-1.0	-4.7	-1.9	-1.6	-1.2	-3.5	-2.2
	BQSquare	-2.2	-29.3	-28.9	-8.6	-2.8	-10.0	-7.5
	BlowingBubbles	-1.9	-10.0	-11.4	-5.7	-2.5	-6.6	-5.3
	RaceHorses	-1.1	-2.8	-2.5	-1.8	-1.2	-2.1	-1.9
	Average	-1.5	-11.7	-11.2	-4.4	-1.9	-5.6	-4.2
Class E WQVGA	Vidyo1	-10.1	-10.7	-10.3	-10.2	-10.0	-10.1	-10.1
	Vidyo3	-10.4	1.6	-5.3	-9.0	-10.0	-5.3	-8.0
	Vidyo4	-10.4	-11.1	-9.9	-10.3	-10.3	-10.7	-10.5
	Average	-10.3	-6.7	-8.5	-9.8	-10.1	-8.7	-9.5
All	Average	-3.9	-9.1	-9.9	-6.0	-4.2	-6.5	-5.8

- **Collect color system metadata for test sequences**
- **Compute R'G'B' PSNR in addition**
 - May want to use weighted average of MSE for R', G', B'
 - We seek input on appropriate set of weights
- **Modify HM to compute PSNR instead of APSNR**