

## Transforms between luma coefficient sets

Charles Poynton,  
*A Technical Introduction  
 to Digital Video*. New York:  
 John Wiley & Sons, 1996.

This note presents matrix transformations that encode and decode between gamma-corrected (nonlinear)  $R'G'B'$  and  $Y'P_B P_R$  components for different luma coefficients. I assume that you are familiar with chapters 5 through 9 of my book, and that you are aware of the ambiguity and confusion in television engineering surrounding the term *luminance*. I also assume that you have read the companion document, *The magnitude of nonconstant luminance errors*.

I express signal processing operations in Mathematica-like pseudocode. The midline dot symbol ( $\cdot$ ) represents the dot product operation. Each transform is represented by a  $3 \times 3$  matrix. To accomplish a transform, place the row vector to be transformed to the right of the matrix, then perform the matrix multiplication.

This matrix codes SMPTE 240M  $R'G'B'$  into SMPTE 240M  $Y'P_B P_R$ :

Eq 1

$$\begin{aligned} \text{encode}_{240M} &= Y'P_B P_R \text{RGB}[Luma_{240M}, \{0, 0, 0\}, \{1, .5, .5\}] \\ & \begin{array}{ccc} 0.212 & 0.701 & 0.087 \\ -0.116101 & -0.383899 & 0.5 \\ 0.5 & -0.444797 & -0.055203 \end{array} \end{aligned}$$

This matrix decodes Rec. 601  $Y'P_B P_R$  into Rec. 601  $R'G'B'$ :

Eq 2

$$\begin{aligned} \text{decode}_{601} &= \text{Inverse}[Y'P_B P_R \text{RGB}[Luma_{601}, \{0, 0, 0\}, \{1, .5, .5\}]] \\ & \begin{array}{ccc} 1. & 0. & 1.402 \\ 1. & -0.344136 & 0.714136 \\ 1. & 1.772 & 0. \end{array} \end{aligned}$$

The concatenation of these matrices can be applied to nonlinear  $R'G'B'$ , to obtain the same effect as SMPTE 240M  $Y'P_B P_R$  encoding followed by Rec. 601  $R'G'B'$  decoding. This is the effect on  $R'G'B'$

components of a downconverter from SMPTE 240M HTDV to 525/59.94 that performs no luma/color difference compensation:

Eq 3

$$\text{decode}_{601} \cdot \text{encode}_{240M}$$

$$\begin{bmatrix} 0.913 & 0.077395 & 0.009605 \\ -0.105114 & 1.150759 & -0.045646 \\ 0.006269 & 0.020731 & 0.973 \end{bmatrix}$$

You can see that red is attenuated by about 9 percent; green is boosted by 15 percent. Here is the analogous matrix for Rec. 709  $Y'P_B P_R$  encoding followed by Rec. 601  $R'G'B'$  decoding:

Eq 4

$$\text{decode}_{601} \cdot Y'P_B P_R \text{RGB}[Luma_{709}, \{0, 0, 0\}, \{1, .5, .5\}]$$

$$\begin{bmatrix} 0.9135 & 0.07858 & 0.00792 \\ -0.105163 & 1.172439 & -0.067277 \\ 0.009596 & 0.032304 & 0.9581 \end{bmatrix}$$

Here is the effect on nonlinear  $R'G'B'$  of Rec. 601 decoding followed by SMPTE 240M  $Y'P_B P_R$  interpretation. This is the effect of a 525/59.94 upconverter that has no luma/color difference compensation:

Eq 5

$$\text{Inverse}[\text{encode}_{240M}] \cdot \text{Inverse}[\text{decode}_{601}]$$

$$\begin{bmatrix} 1.087 & -0.072852 & -0.014148 \\ 0.098928 & 0.861627 & 0.039444 \\ -0.009112 & -0.017888 & 1.027 \end{bmatrix}$$

Here is the analogous matrix for Rec. 709:

Eq 6

$$\text{Inverse}[Y'P_B P_R \text{RGB}[Luma_{709}, \{0, 0, 0\}, \{1, .5, .5\}]] \cdot \text{Inverse}[\text{decode}_{601}]$$

$$\begin{bmatrix} 1.0865 & -0.072433 & -0.014067 \\ 0.096643 & 0.844833 & 0.058524 \\ -0.01414 & -0.02776 & 1.0419 \end{bmatrix}$$

I assert that no significant performance benefit would be obtained by choosing, for HDTV, luma coefficients that theoretically match the chosen HDTV primaries. The reason that there would be no significant benefit is that the right numbers would still be used in the wrong equation! Color science provides the "correct" coefficients for a calculation performed in tristimulus (linear light) values, but in video, we perform the calculation on nonlinear (gamma corrected) values instead!

On the other hand, huge penalties await if we choose, for HDTV, luma coefficients that are different from the ubiquitous Rec. 601 coefficients used in all forms of digital and analog video today. The penalties would be in two areas. If we choose to include luma and color difference processing every time we upconvert or downconvert, the penalty involves a huge cost - the matrix multiplication operations in this note have to be implemented. If we choose to ignore the issue, then errors of a magnitude indicated by the matrices above will be introduced into our signals every time we upconvert or downconvert.