

Supplement to Chapter 2 of *The Science of Digital Media* – Digital Image Representation

Programming Exercise – Digital Imaging > Converting Between Color Models¹

Introduction:

Color is represented in a variety of ways in image processing environments. RGB is a commonly-used color model that separates color into three primary components. This model runs parallel to the way pixels are displayed on a computer monitor, with three color cells per pixel. CMYK is the model used in professional ink-printing processes. HSV and HLS separate color into hue, saturation, and brightness components. Luminance/chrominance models like YCbCr, YIQ, and YUV separate color into one luminance and two chrominance components.

Translating color information from one model to another is sometimes a linear transformation. This is the case when going from RGB to YIQ, as defined by the matrix operation below.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Translating from RGB to CMYK involves the following steps:

```
Cinit = 1-R  
Minit = 1-G  
Yinit = 1-B  
K = minimum(Cinit, Minit, Yinit)  
C = Cinit-K  
M = Minit-K  
Y = Yinit-K
```

RGB to HSV conversion is a non-linear transformation that uses the algorithm below:

```
algorithm RGB_to_HSV  
/* Input: r, g, and b, each real numbers in the range [0..1].  
Output: h, a real number in the range of [0..360), except if s = 0, in which case h is  
undefined. s and v are real numbers in the range of [0..1].*/  
{  
    max = maximum(r,g,b)  
    min = minimum(r,g,b)
```

¹This material is based on work supported by the National Science Foundation under Grant No. DUE-0340969. This worksheet was written by Jennifer Burg.

```
v = max
if max ≠ 0 then s = (max - min)/max
else s = 0
if s == 0 then h = undefined
else {
  diff = max - min
  if r == max then h = (g - b) / diff
  else if g == max then h = 2 + (b - r) / diff
  else if b == max then h = 4 + (r - g) / diff
  h = h * 60
  if h < 0 then h = h + 360
}
```

RGB to HLS conversion is also a non-linear transformation and uses this algorithm:

```
algorithm RGB_to_HLS
/* Input r, g, and b, each real numbers in the range [0...1]
Output: h, a real number in the range of [0...360], except if s = 0, in which case h is
undefined.. t and s are real numbers in the range of [0...1].*/
{
  max = maximum(r,g,b)
  min = minimum(r,g,b)
  t = average(max, min)
  if max == min then s = 0
  else {
    sum = max + min
    diff = max - min
    if t ≤ 0.5 then s = diff / sum
    else s = diff / (2 - max + min)
    r_temp = (max - r) / diff
    g_temp = (max - g) / diff
    b_temp = (max - b) / diff
    if r == max then h = b_temp - g_temp
    else if g == max then h = 2 + r_temp - b_temp
    else if b == max then h = 4 + g_temp - r_temp
    h = h * 60
    if h < 0 then h = h + 360
  }
}
```

Instructions:

The Assignment

Using the programming language of your choice, implement conversions from RGB to YIQ

RGB to CMYK
RGB to HSV
RGB to HLS

You can write your program so that it either prompts the user for RGB values or reads them from a file.

You may want to include a menu at the beginning which asks the user which type of conversion he or she wants to do.

Ideas for Further Experimentation

- Write and implement algorithms for HLS to RGB and/or HSV to RGB.
- Compare the answers you get from the RGB to HSV and RGB to CMYK conversions with what you get in Photoshop. You can do this by opening the color chooser in Photoshop and examining the values there. Notice that the Photoshop values for CMYK don't match the conversion given here. See if you can determine the method Photoshop uses for the RGB to CMYK conversion.