

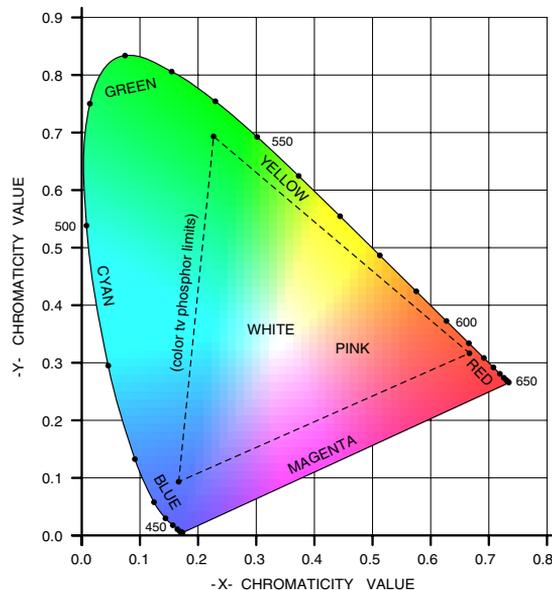
# Some Possible "False Color" and "Rainbow" Improvements

**Don Lancaster**  
**Synergetics, Box 809, Thatcher, AZ 85552**  
copyright c2007 as **GuruGram #82**  
<http://www.tinaja.com>  
[don@tinaja.com](mailto:don@tinaja.com)  
(928) 428-4073

I often use extensive "real" color in my tech illustrations, [web presentations](#), and my .PDF files. Such as this [CIE Chromaticity Diagram](#), or this [NTSC Color Phase Wheel](#). I also use "false color" to illustrate such items as these [Electromagnetic Field Plots](#), or this [Three Phase Animation Demo](#).

Typically, any simple or obvious "rainbow" presentations end up with "**The red is too dark and the blue is waay too dark**". And "**Too much green, not enough orange or yellow**". I seemed to be unable to find these problems addressed anywhere obvious, so I thought we might look at them here.

Lets first review this [CIE Chromaticity Diagram](#) that shows how the colors, color wavelengths, and common phosphors relate to each other...



Your usual rainbow runs red through orange, yellow, green, cyan (or aqua), blue, and violet. Colors near "magenta" are problematical for rainbow or false color use because they take a "short cut" across the bottom of the chromaticity diagram. They are also **not** "real" colors in that they consist mostly of white minus various amounts of green. And typically require fractional blending of red and blue.

Thus a useful rainbow or false color set might range from red through green to blue. Alternately, we might like to reverse things. Since red is often perceived as "hot" and blue as "cool".

One easy way to explore color and rainbow variations is with the general purpose **PostScript** computer language. But your final result can be a simple **color table lookup** that can be used with most any language or application package on nearly any system. Details on the actual **PostScript** code used here appear in the **sourcecode** to this **GuruGram**.

For rainbow or false color use, the **HSB** or **hue-saturation-brightness** color space likely will prove the most useful for exploration. HSB **fairly easily 1:1 maps** to the more conventional **RGB** color space.

In the PostScript HSB color space, the hue varies from **0.00** for red through **0.33** for green through **0.67** for blue. A color is white at **0.0** and fully saturated at **1.0**. Brightness is black at **0.0** and maximum at **1.0**.

Our chromaticity diagram shows us that colors vary from red with a wavelength of **650** nanometers up through blue with a wavelength of **450** nanometers. With the **shorter** wavelengths corresponding to **higher** optical frequencies.

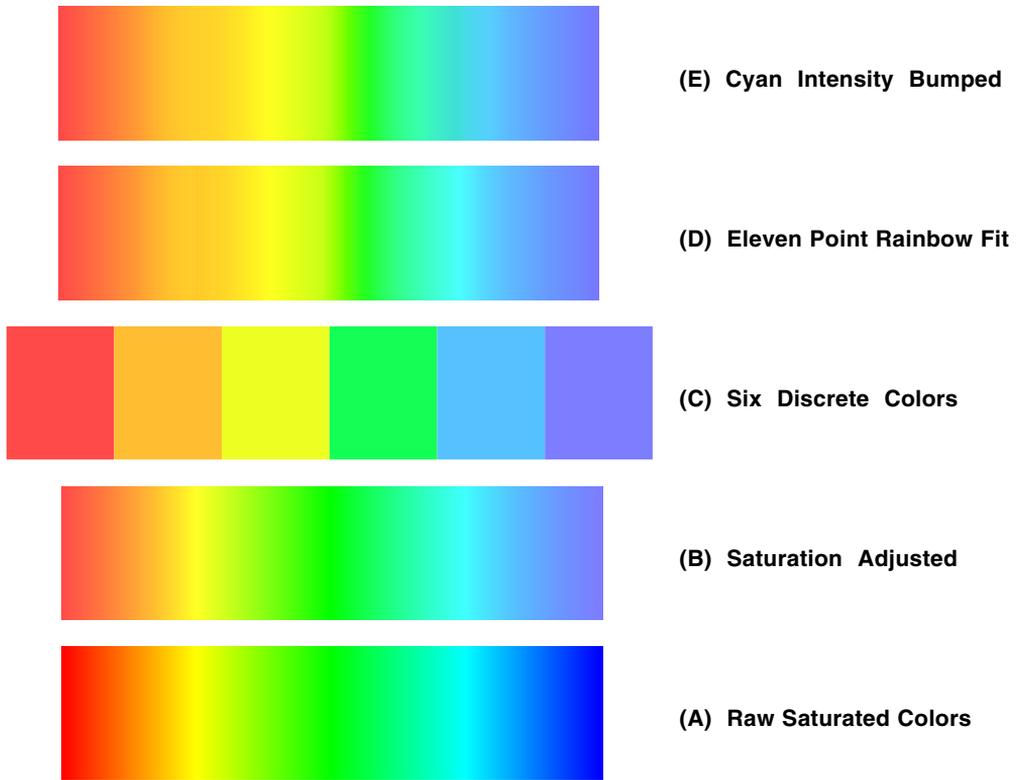
Fully saturated colors are along the edge of the curve. And decrease in saturation to white near the middle. Brightness varies "out of the page". Thus, all the colors can be represented by a **stack** of chromaticity diagrams, one on top of the other. With black on the bottom and brightest nearest you.

**No printing or display process is known that can show all colors to any and all saturations.** Thus some degree of compromise is essential. Color tv phosphors in particular limit you to the colors inside the triangle shown. These phosphors are red, blue, and green. Oranges and yellows result from mixing red and green. Aquas or cyans result from mixing green and blue. While magenta is a mix of red and blue, it can also be thought of white from which varying amounts of green are removed.

Less saturated colors result from mixing the odd phosphor out. For instance, orange will be a mix of red and green. Adding varying amounts of blue will move you from orange (no blue) through white (lots of blue) as you unsaturate.

## Getting Started

Let's pre-gather all of our upcoming rainbow improvement results together here in one place so you can easily compare their progress...



We will start at the bottom of the stack and work our way up. A simple and obvious first attempt at a rainbow is to vary the hue from **0** to **0.667** while holding the saturation at **1.0** and the brightness at **1.0**. Which results in plot **(A)** and is typical of many initial rainbow or false color attempts.

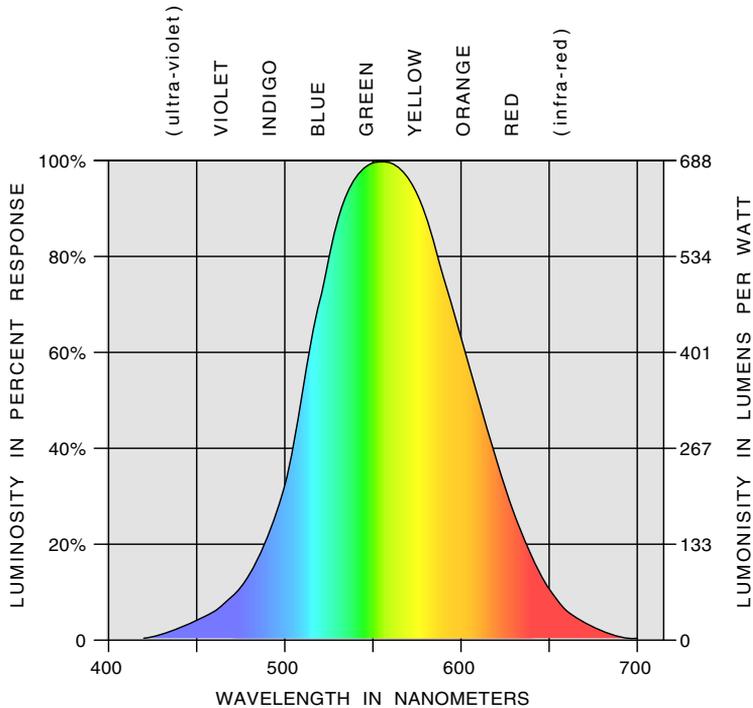
We do end up with a rainbow all right, except it looks "wrong" somehow. The reds are somewhat too dark. The blues are waaaay too dark. There is too much green, and not enough yellow or orange. These problems are caused by the human eye sensitivity being highest in the green, lesser in the red, and rather weak in the blue. Combined with the eye's log color sensitivity.

We'll note in passing that "wider" rainbows will have fewer glitch artifacts than "narrow" ones. Especially on lower resolution screen presentations. A **three inch minimum** is probably a good idea with conventional VGA screen displays. Any artifacts should sharply diminish with much higher print resolutions.

We are going to need as many colors as we can get, so as many of the 12-bit RGB colors as possible should be used. (A) and (B) are shown with one hundred colors, while (D) and (E) are shown with two hundred. Any attempt to limit yourself to the 64 web friendly colors is likely to give abysmal results.

## Adjusting Saturation

Here is how the "standard eyeball" sensitivity varies with color...



We see that the eye is quite good at seeing green, mediocre at seeing red, and rather poor at seeing blue. Thus, if we have three colors of equal energy, the eye will see a darker red and a much darker blue.

There are formulas for the equivalent equal brightness gray for any color. One widely used formula applies to NTSC television color phosphors, the PostScript color conventions, and most newer monitors...

$$\text{equivalent gray} = 0.29 \text{ red} + 0.59 \text{ green} + 0.11 \text{ blue}$$

This formula is normally used to make sure a color presentation will also print in an expected manner in black and white. But we can apply it here to "soften" the harshness of our darker red and blue colors.

Chances are we would like to keep our greens fully saturated. For equivalent grays and equal eyeball perception, we should **desaturate** the reds and blues.

An algorithm I've found useful: For a pure red, back off the saturation by  $0.59 - 0.30 = 0.29$ . Changing the saturation from  $1 - 0.29 = 0.71$ . For a pure blue, you back off the saturation by  $0.59 - 0.11 = 0.48$ . And changing your saturation from  $1$  to  $1 - 0.48 = 0.52$ .

For intermediate hue values, you proportion the change to the fraction of red to green or the fraction of blue to green present. Thusly...

#### TO DESATURATE FOR EQUAL PERCEPTION...

For red through green hues of less than 0.33...

$$\text{sat} = 1 - (0.29 - 0.879 * \text{hue})$$

For green through blue hues of 0.33 to 0.67...

$$\text{sat} = 1 - (1.455 * (\text{hue} - 0.33))$$

Note that  $0.879$  is a slope of  $0.29/0.33$  and  $1.45$  is a slope of  $0.48/0.33$ . Note further that the green correction ends up as zero.

Here is the algorithm done in **PostScript**...

```
curhue 0.33 lt
  {1 0.29 curhue 0.879 mul sub sub }
  {1 curhue 0.33 sub 0.48 0.33 div mul sub } ifelse
```

Rainbow **(B)** above shows this adjustment.

## Improving Color Linearity

Our rainbow **(B)** is obviously greatly improved. But we still seem to have far too much green and not nearly enough in the way of oranges or yellows. It turns out that the eye has a somewhat **logarithmic** response to color perception. And is able to resolve the **lower** optical frequencies into more perceived color differences.

Our approach to evening out the colors will be to create a **lookup table** that relates the actual hue to the hue we really want to see.

Let's assume we have a small child's box of crayons and wish to extract **six** colors of already corrected saturations. We would like to properly spread out the colors and have as strong a difference between each color as possible.

Selecting these colors can be highly subjective, but one possible color set appears as **(C)** above. And gives us red, orange, yellow, green, blue, and a violet.

We then attempt to create a **hue adjustment lookup table** that gives an eleven point fit to these six colors. The **center** of each color should exactly match. The **edge** between adjacent colors should "split the difference" favoring neither.

In general, most hues get reduced, favoring the oranges and yellows. But a few blues actually will get raised slightly. Since the eye is ultra-sensitive to any nuance, **your correction table should have no sudden slope changes** in it. A simple running seven point average filtering will usually handle this.

The hue corrected table lookup combines saturation adjustment with hue correction in one step. You can view the results as rainbow (**D**). To use it, you intercept your input hue and change it to your desired hue.

Here is one possible **PostScript** rainbow hue and saturation correction table...

```
/nicerainbowdata [
```

```
[0.000 0.710 1.0][0.002 0.712 1.0][0.006 0.715 1.0][0.008 0.718 1.0][0.012 0.721 1.0][0.015 0.725 1.0][0.018 0.727 1.0][0.021 0.731 1.0]
[0.024 0.733 1.0][0.027 0.737 1.0][0.030 0.739 1.0][0.033 0.743 1.0][0.036 0.745 1.0][0.039 0.749 1.0][0.042 0.751 1.0][0.045 0.755 1.0]
[0.048 0.757 1.0][0.050 0.760 1.0][0.054 0.763 1.0][0.056 0.766 1.0][0.060 0.769 1.0][0.062 0.772 1.0][0.066 0.775 1.0][0.068 0.778 1.0]
[0.072 0.781 1.0][0.074 0.784 1.0][0.078 0.787 1.0][0.080 0.790 1.0][0.084 0.793 1.0][0.086 0.796 1.0][0.090 0.799 1.0][0.093 0.802 1.0]
[0.096 0.805 1.0][0.099 0.808 1.0][0.102 0.811 1.0][0.105 0.814 1.0][0.107 0.816 1.0][0.110 0.820 1.0][0.112 0.821 1.0][0.115 0.824 1.0]
[0.116 0.825 1.0][0.119 0.828 1.0][0.119 0.828 1.0][0.122 0.832 1.0][0.122 0.831 1.0][0.125 0.834 1.0][0.124 0.833 1.0][0.127 0.836 1.0]
[0.125 0.834 1.0][0.128 0.836 1.0][0.124 0.836 1.0][0.130 0.839 1.0][0.128 0.837 1.0][0.131 0.840 1.0][0.129 0.838 1.0][0.132 0.842 1.0]
[0.131 0.840 1.0][0.134 0.843 1.0][0.133 0.842 1.0][0.136 0.845 1.0][0.135 0.845 1.0][0.139 0.844 1.0][0.138 0.847 1.0][0.144 0.851 1.0]
[0.141 0.851 1.0][0.145 0.854 1.0][0.145 0.854 1.0][0.148 0.858 1.0][0.149 0.858 1.0][0.152 0.861 1.0][0.154 0.862 1.0][0.156 0.865 1.0]
[0.157 0.866 1.0][0.160 0.869 1.0][0.161 0.870 1.0][0.164 0.873 1.0][0.165 0.874 1.0][0.168 0.877 1.0][0.169 0.878 1.0][0.172 0.881 1.0]
[0.173 0.882 1.0][0.176 0.885 1.0][0.177 0.886 1.0][0.180 0.889 1.0][0.181 0.890 1.0][0.184 0.894 1.0][0.185 0.894 1.0][0.188 0.897 1.0]
[0.189 0.898 1.0][0.192 0.901 1.0][0.193 0.902 1.0][0.196 0.905 1.0][0.197 0.906 1.0][0.200 0.909 1.0][0.201 0.910 1.0][0.204 0.913 1.0]
[0.206 0.915 1.0][0.209 0.918 1.0][0.214 0.923 1.0][0.218 0.926 1.0][0.225 0.935 1.0][0.228 0.931 1.0][0.238 0.983 1.0][0.242 0.978 1.0]
[0.254 0.965 1.0][0.257 1.000 1.0][0.271 1.000 1.0][0.274 1.000 1.0][0.289 0.928 1.0][0.292 0.925 1.0][0.307 0.909 1.0][0.311 0.906 1.0]
[0.324 0.891 1.0][0.328 0.888 1.0][0.340 0.875 1.0][0.343 0.872 1.0][0.353 0.860 1.0][0.357 0.857 1.0][0.366 0.846 1.0][0.370 0.844 1.0]
[0.378 0.836 1.0][0.381 0.835 1.0][0.389 0.824 1.0][0.392 0.821 1.0][0.398 0.814 1.0][0.402 0.811 1.0][0.402 0.805 1.0][0.411 0.802 1.0]
[0.416 0.796 1.0][0.419 0.792 1.0][0.424 0.787 1.0][0.428 0.784 1.0][0.433 0.778 1.0][0.436 0.775 1.0][0.441 0.770 1.0][0.445 0.762 1.0]
[0.450 0.761 1.0][0.453 0.757 1.0][0.458 0.757 1.0][0.461 0.749 1.0][0.466 0.743 1.0][0.470 0.740 1.0][0.475 0.734 1.0][0.478 0.731 1.0]
[0.484 0.725 1.0][0.487 0.722 1.0][0.492 0.717 1.0][0.495 0.713 1.0][0.501 0.708 1.0][0.504 0.704 1.0][0.509 0.699 1.0][0.512 0.695 1.0]
[0.518 0.693 1.0][0.521 0.687 1.0][0.526 0.681 1.0][0.529 0.678 1.0][0.534 0.673 1.0][0.537 0.669 1.0][0.542 0.665 1.0][0.545 0.661 1.0]
[0.549 0.657 1.0][0.552 0.654 1.0][0.556 0.650 1.0][0.559 0.647 1.0][0.562 0.643 1.0][0.566 0.640 1.0][0.569 0.637 1.0][0.572 0.633 1.0]
[0.574 0.631 1.0][0.578 0.627 1.0][0.580 0.625 1.0][0.583 0.621 1.0][0.586 0.619 1.0][0.589 0.615 1.0][0.592 0.612 1.0][0.595 0.609 1.0]
[0.597 0.607 1.0][0.601 0.603 1.0][0.603 0.601 1.0][0.607 0.597 1.0][0.609 0.595 1.0][0.614 0.591 1.0][0.615 0.589 1.0][0.618 0.585 1.0]
[0.621 0.583 1.0][0.624 0.579 1.0][0.626 0.577 1.0][0.630 0.573 1.0][0.632 0.575 1.0][0.635 0.567 1.0][0.638 0.565 1.0][0.641 0.561 1.0]
[0.644 0.559 1.0][0.643 0.553 1.0][0.649 0.553 1.0][0.653 0.549 1.0][0.655 0.547 1.0][0.658 0.543 1.0][0.661 0.541 1.0][0.664 0.537 1.0]
[0.667 0.535 1.0]
```

```
] store
```

This lookup table has 201 entries. To use it, you convert your input hue to an integer from 0 to 200. Then you read and use the equivalent hue, saturation, and brightness. For a hundred table hues, just use every second entry. Alternately, an equivalent table can be created with RGB values.

You can magnify these table values to read them more easily. Or you can extract the tables directly from the [sourcecode](#) to this [GuruGram](#).

## Gilding the Lily

By using a table lookup, we can dramatically improve the presentations of rainbows and false color. As **(D)** above clearly shows us. We still seem to have aqua or cyan being "a little hot". The cause of this may be differences in individual perception, phosphor performance, backlighting spectral peaks or monitor setting variations.

I know of no particular physical or perception reason why cyan should be hotter here. **Any further "improvements" might run afoul of individuals and their viewing systems.** Nonetheless, we have possibly "improved" our rainbow **(E)** by backing off on the brightness in and around cyan.

We ramp into and out of a maximum brightness reduction of around **0.12**. The present changes to our lookup table are...

```

. . . . .
[0.436 0.775 1.0 0.01 sub ] [0.441 0.770 1.0 0.02 sub ]
[0.445 0.764 1.0 0.03 sub ] [0.450 0.761 1.0 0.04 sub ]
[0.453 0.757 1.0 0.05 sub ] [0.458 0.752 1.0 0.06 sub ]
[0.461 0.749 1.0 0.07 sub ] [0.466 0.743 1.0 0.08 sub ]
[0.470 0.740 1.0 0.09 sub ] [0.475 0.734 1.0 0.10 sub ]
[0.478 0.732 1.0 0.11 sub ] [0.484 0.725 1.0 0.12 sub ]
[0.487 0.722 1.0 0.12 sub ] [0.495 0.717 1.0 0.13 sub ]
[0.495 0.713 1.0 0.12 sub ] [0.501 0.708 1.0 0.12 sub ]
[0.504 0.704 1.0 0.11 sub ] [0.509 0.699 1.0 0.10 sub ]
[0.512 0.695 1.0 0.09 sub ] [0.518 0.690 1.0 0.08 sub ]
[0.521 0.687 1.0 0.07 sub ] [0.526 0.681 1.0 0.06 sub ]
[0.529 0.678 1.0 0.05 sub ] [0.534 0.673 1.0 0.04 sub ]
[0.537 0.669 1.0 0.02 sub ] [0.542 0.665 1.0 0.01 sub ]
. . . . .
```

This cyan bumping table is not quite optimum yet in that a minor hotspot and some roughness remains. Fixing these is left as an exercise for the reader.

## **For More Help**

Similar tutorials and additional support materials are found on our **Acrobat**, our **PostScript** and our **GurGram** library pages. As always, **Custom Consulting** is available on a cash and carry or contract basis. As are seminars.

For details, you can email [don@tinaja.com](mailto:don@tinaja.com). Or call **(928) 428-4073**.