

Further developing a colour mark similarity measurement framework – Part III: A method for sorting colours

by David Barnett

Introduction

In my previous articles^{1,2,3} looking at a framework for analysis of colour marks, I considered the use of the RGB definition for colours (specifying their individual red, green and blue components each as integer values between 0 and 255), and how this can be used to specify the ‘distance’ (d) between any two colours⁴ and, equivalently, a similarity score (S_{col})⁵ for the pair.

However, when considering colour marks, it can also be helpful to have an algorithm for *sorting* a list of colours into a convenient order for visual review. The obvious choice would be an ordering which resembles a *spectrum* of colours. This is distinct from a simple ordering based on just a sorting of (say) the numerical R values, followed by the G values and then the B values (as per Figure 1 in the previous article), which generates multiple, near-repeating series of coloured ‘bands’.

The difficulty is that there is no simple algorithm for translating a 3D colour space into a (1D) linear series of colours in which all transitions are smooth and continuous.

The situation can be appreciated by visualising the same set of 4,096 representative, ‘regularly-spaced’ colours as considered in the previous article (i.e. [8,8,8], [8,8,24], [8,8,40], ... , [8,24,8], [8,24,24], [8,24,40], ... , [24,8,8], ... , [248,248,248]).

Algorithms for sorting colours are frequently based on expression of the colours in HSV, rather than RGB, format⁶. This alternative representation also uses three components:

- **H (hue)** – the ‘base’ colour (on a scale from 0 to 1 in ‘spectral’ order)
- **S (saturation)** – the intensity of the colour
- **V (‘value’)** – the darkness of the colour

Mathematical conversion of the RGB expression of a colour to its HSV equivalent involves a simple algorithm, and a number of pre-written library scripts⁷ are available to implement it.

¹ <https://circleid.com/posts/towards-a-quantitative-approach-for-objectively-measuring-the-similarity-of-marks>

² <https://circleid.com/posts/further-developing-a-colour-mark-similarity-measurement-framework-building-a-database>

³ ‘Further developing a colour mark similarity measurement framework – Part II: Defining a similarity score’

⁴ $d = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$

⁵ $S_{col} = 1 - [d / \sqrt{3 \times 255^2}]$

⁶ <https://www.alanzucconi.com/2015/09/30/colour-sorting/>

⁷ e.g. <https://github.com/python/cpython/blob/3.13/Lib/colors.py>

```
def rgb_to_hsv(r, g, b):
    maxc = max(r, g, b)
    minc = min(r, g, b)
    rangec = (maxc - minc)
    v = maxc
    if minc == maxc:
        return 0.0, 0.0, v
    s = rangec / maxc
    rc = (maxc - r) / rangec
    gc = (maxc - g) / rangec
    bc = (maxc - b) / rangec
    if r == maxc:
```

The simplest method of sorting colours is just straightforwardly by their H (hue) values. For the set of 4,096 colours considered previously, this gives an ordering as shown in Figure 1 (left to right, then top to bottom).

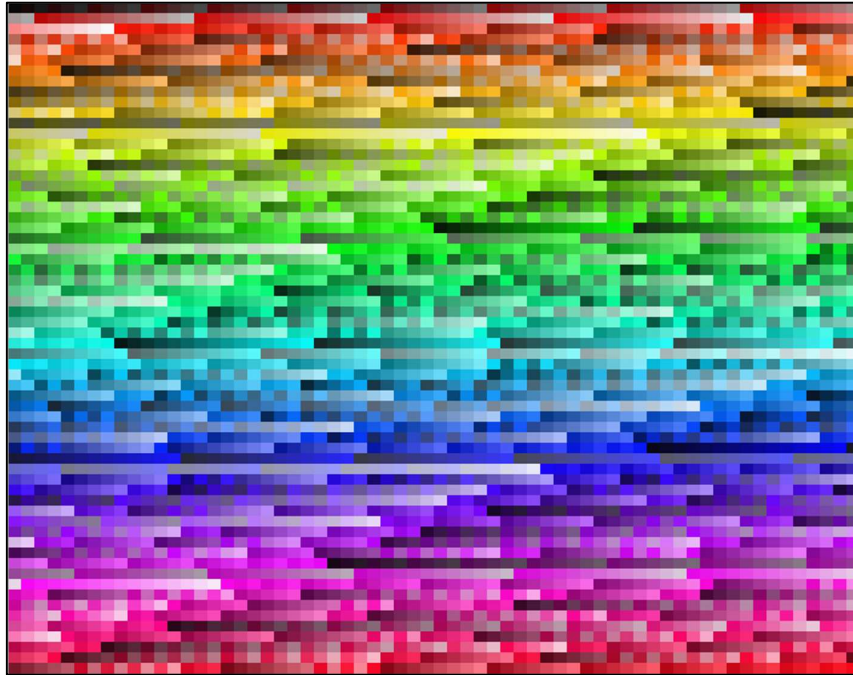


Figure 1: Ordering of 4,096 colours occupying regularly-spaced positions in RGB space, according to their H (hue) values

What is *less* satisfactory about this ordering is that it takes no account of the other two parameters, so there are (for example) rapid alternations between dark and light shades, but there is no way to entirely smooth out these discontinuities without losing the smoothness of the transitions according to the other parameter(s).

One other option is the use of an additional parameter, **L (luminosity)**. (Perceived) luminosity can be derived directly from the RGB values of a colour⁸; on its own, it does not provide a good basis for sorting colours, but can be combined with the use of H to provide smoother transitions. One such option is to divide the H values into 'blocks' and then sort by L within each block. However, this still results in sharp transitions between adjacent colours at the *ends* of blocks, so does not really add much value in many cases. In the remainder of this article, therefore, sorting by (just) the H parameter is utilised.

```
h = bc-gc
elif g == maxc:
    h = 2.0+rc-bc
else:
    h = 4.0+gc-rc
h = (h/6.0) % 1.0
return h, s, v
```

⁸ $L = \sqrt{0.241 \times R + 0.691 \times G + 0.068 \times B}$

Applications of colour sorting / ordering

The first point to note is that the H value (i.e. the position of the colour in an ordered spectrum) does not *in itself* provide the basis for an effective metric for comparing the similarity of colours (compared with the geometric distance (d) in colour space discussed previously), in part due to the disregarding of the other two components which affect a colour's visual appearance.

As a related point, the relationship between hue (H) and colour distances (d) is complex, due to the distribution of colours in 3D space. As one illustration of this, it is instructive to visualise the numerical distance (d) of each of the 4,096 colours shown in Figure 1 from a *fixed* colour, as a function of the H value of the individual (*variable*) colour in each case. This relationship is shown in Figure 2, for *three* fixed colours: pure red ([255,0,0]), pure green ([0,255,0]) and pure blue ([0,0,255]).

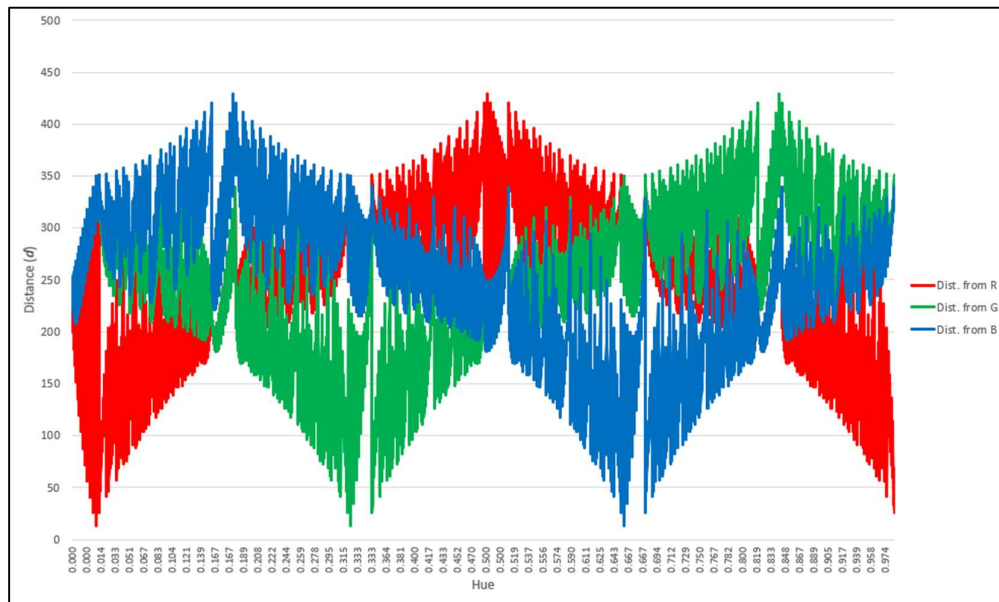


Figure 2: Distances (d) of each of the 4,096 colours in Figure 1 from (pure) red, green and blue, as a function of their H (hue) values (0 = red; 1 = violet)

Figure 2 also reveals the 'circular' nature of the colour spectrum (when ordered according to hue), with both the 'red' and 'violet' ends of the spectrum 'close' to 'pure' red (i.e. [255,0,0]) – another reason why hue (H) is a less satisfactory basis (than d) for quantifying the proximity of colour pairs.

However, there are practical uses for sorting by H, predominantly where it is useful to be able to visually review sets of colours as part of the analysis process for marks (e.g. where assessing disputes and potential colour 'clashes').

For example, if maintaining a database of registered colour marks, it may be useful to have them sorted into a meaningful order, to be able visually review the proximity of similar marks and determine whether new proposed colour-marks are close enough to others to present a potential problem. For example, the set of colour marks considered in the 'Building a database' article in this series is again presented in Table 1, but here with the colours sorted by their H values, providing a much more preferable basis for manual review. (The table also illustrates how visually 'darker' shades are, in general, associated with lower L values.)
























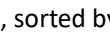
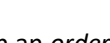
Brand	R	G	B	Colour	H	L
Coca Cola	244	0	0		0.000	7.7
Target	204	0	0		0.000	7.0
Transport for London	212	46	18		0.024	9.2
UPS	53	28	21		0.036	5.8
Reese's	254	82	0		0.054	10.9
Home Depot	250	99	4		0.064	11.4
University of Texas	191	87	0		0.076	10.3
Orange Mobile	255	121	0		0.079	12.0
Fiskars	255	142	0		0.093	12.6
Veuve Clicquot	239	162	32		0.105	13.1
Qualitex	216	187	120		0.116	13.8
Caterpillar	255	197	0		0.129	14.1
Wiffle Ball	250	240	95		0.156	15.2
Post-It	255	255	153		0.167	15.8
Slip 'N Slide	255	255	0		0.167	15.4
John Deere	54	124	43		0.311	10.1
Tiffany	129	216	208		0.485	13.9
University of North Carolina	75	156	211		0.567	11.8
Milka	104	79	163		0.716	9.5
Cadbury's	51	0	114		0.741	4.5
Barbie	224	33	138		0.908	9.3
T-Mobile	226	0	116		0.914	7.9
Owens-Corning	249	159	201		0.922	13.5
Harvard	164	16	52		0.959	7.4
Louboutin	238	31	37		0.995	9.0

Table 1: Mock-up of a database of protected colour marks, sorted by their H values

In a second application, it might be helpful to be able to visualise (in an *ordered* form) the *set* of colours which are similar to a *particular degree* to another *fixed* colour, building on the idea of the similarity score (S_{col}) presented in the previous article.

For example, taking an arbitrary colour somewhere near the centre of the colour space (say, [136,72,56], a shade of brown; Figure 3), it might be instructive to be able to visualise the set of colours (and the extent of their variability!) which would be deemed (by S_{col}) as being (for example) 75% similar (i.e. those colours sitting on the surface of a sphere in RGB space of appropriate radius – in this case, $d = 110$ RGB units – surrounding the colour in question). Such analyses might be informative in formulating guidelines regarding the thresholds up to which colour-mark protection might apply. Figure 4 shows the range of such colours, again sorted by H values, taken from the dataset of 4,096 colours considered previously. The examples range from [40,24,24], [232,24,24] and [232,104,104] (all H = 0.000) to [232,24,40] (H = 0.987).



Figure 3: A rectangle of colour RGB = [136,72,56]

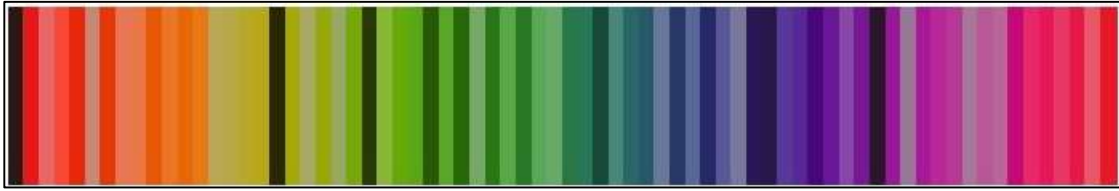


Figure 4: Subset of the group of 4,096 colours occupying regularly-spaced positions in RGB space which are 75% (to the nearest percent) similar (according to S_{col}) to the colour [136,72,56], sorted by their H values

In summary, therefore, whilst the option for sorting colours into a meaningful order does not add much value to the framework for quantifying the *degree of similarity* between colours, it does provide a basis for being able to present colour information in a format which is more easily visually digestible. These ideas therefore could have applications in reviewing dispute cases, selecting options for new potential colour marks, and in formulating guidelines for IP protection thresholds.