Colour Systems

Introduction

All image processing and drawing programs provide a 'colour picker' which enables the user to select any desired colour. Every program seems to employ a different way of presenting these colours and some are definitely not as good as others. Most offer a choice between the simple RGB system and a more sophisticated HSV (hue, saturation, value) system. In this article I wish to explain the relationship between these two systems and explore the possibility that there might be other systems more suited to the task of picking colours.

All the images in this article were produced by a simple program called 'Colour Wheel' which can be downloaded from my website:

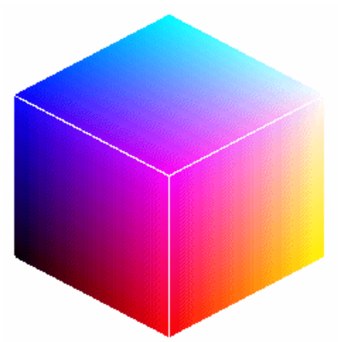
www.btinternet.com/~jolinton/JollySoftware

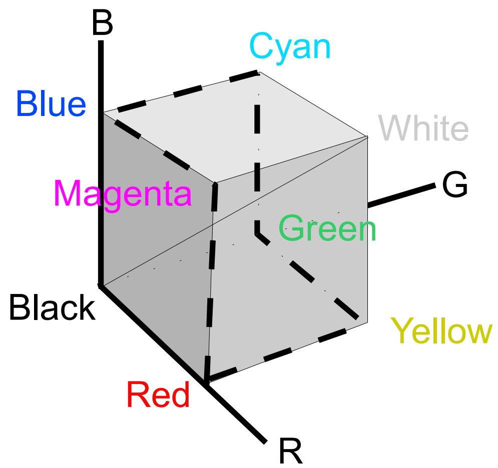
(Please note that these images will not reproduce the full range of colours available as they are .gif images and are restricted to a mere 256 colours.)

The Colour Cube

It is well known that every colour which can be depicted on a colour monitor is defined by three components: red (R), green (G) and blue (B) each of which can take any value from 0 to 255. This enables a staggering number (16,777,216 to be precise) of different colours to be created. (Even so, this is not all of the colours that can be distinguished by the human eye and many colours which have different RGB components look exactly the same as well so the mapping between all the different colours that the human eye can see and those that a computer can generate is not perfect. We shall, however restrict ourselves to the latter set.)

It is convenient to represent each of these 16 million colours as a point within a cube whose three axes are the RGB components like this:

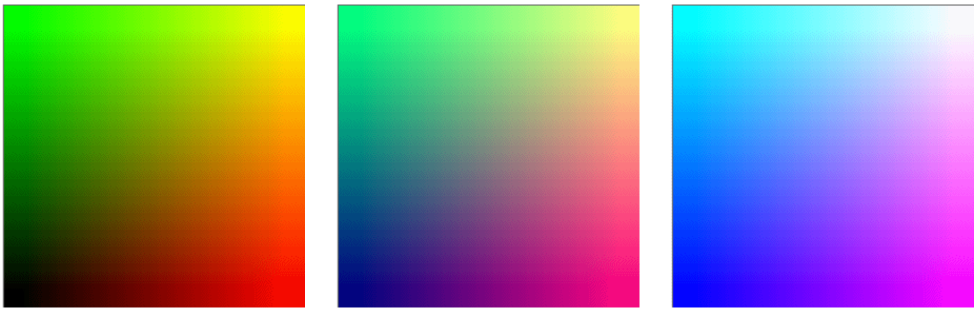




It will be noted that black and white are at the ends of the principal diagonal of the cube while the three primary colours and the three secondary colours (known as **principal** colours) lie on a wiggly line which threads through all the remaining six corners.

When designing a 'colour picker' for use in a painting program, it is impossible to show all the colours of the cube simultaneously on a two dimensional screen so it is usual to show some kind of 'slice' through the cube and to provide a slider which can move the 'slice' up and down. For example, in the RG – B system, the 'slices' are horizontal planes parallel to the RG plane and the slider is the B component. This produces the following types of figure:

The RG – B system



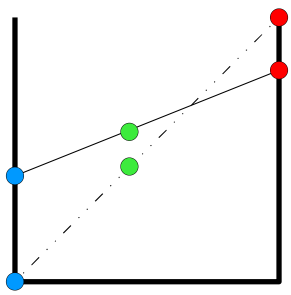
B = 0 B = 128 B = 255

The advantage of this system is that it is immediately clear what is going on and each 'slice' presents an interesting palette of colours from warm to cool.

There is, however, no reason to single out the blue component in preference to the others. We should, therefore also provide two more systems the GB – R system and the BR – G system in which the red and green components can be altered independently. It turns out, however, that the palettes produced by these systems are even more garish and, although in my opinion the RG – B system is the best, it is not easy to find subtle shades like olive drab or pale lilac in any of them. The reason for this is that every point in the square is a totally different colour in that it has a different *balance* of red, green and blue components. What we need is a system in which one of the components represents this balance of primary colours. This component is called **hue**.

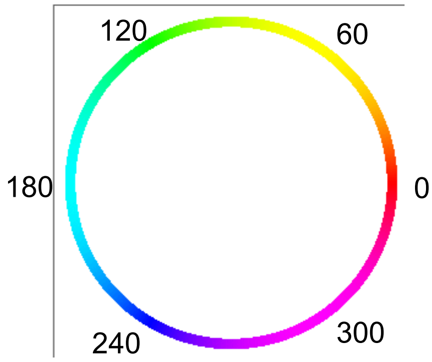
Hue (H)

Base colours have one component equal to 0 and another component equal to 100%. The third component may be anything from 0 to 100%. (100% is represented by the figure 255 in a computer.) All colours of the same **hue** share the same **base colour**. For example, the spots on the solid line in the diagram below represent the RGB components of a certain colour. They appear to be approximately 80%, 50% and 30% respectively. The spots on the dotted line represent the RGB components of the base colour which has the same **hue**. They are 100%, 40% and 0% respectively. The important thing is that the green component stays in the same relative position between the maximum and minimum components.

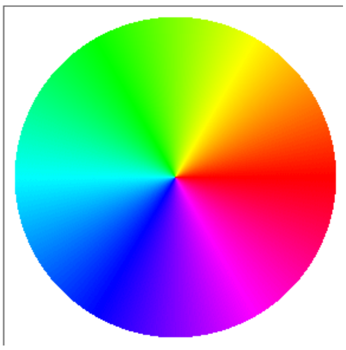


It is of vital importance to realise that adding a constant amount to all three colours (equivalent to 'adding whiteness') does not alter the position of the middle component on the line – it simply shifts the line up and down. i.e. it does not change the **hue**. It is equally important to realise that *multiplying* each component by a constant factor (equivalent to changing the brightness) changes the *gradient* of the line but likewise, it does not change the **hue** either. It follows that *any linear transformation* of the three components only changes the brightness and saturation of the colour – it does not change the hue.

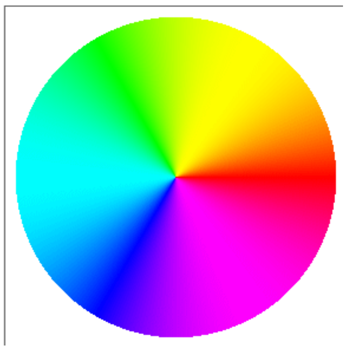
It is easy to see that all the **base colours** lie on the wiggly line which I referred to earlier which threads it way past all the primary and secondary colours. If we could flatten it out we would obtain a hexagon with the primary and secondary colours (ie the principal colours) at the vertices. Better still we can mould it into a '**circle of hues**' like this:



By convention, red is placed at 0^0, green at 120^0 and blue at 240^0. It is usual for the third component to change linearly from one principal colour to the next so that at 30^0, for example, half way between red and yellow, the green component will have the value 128. There is no necessity for this function to be linear, however. In fact, the eye is much more sensitive to small changes around the secondary colours than it is to changes round the primary ones. I have found that a quadratic function spreads the secondary colours and greatly improves the balance of colours round the circle. To see what I mean, compare the following two figures:

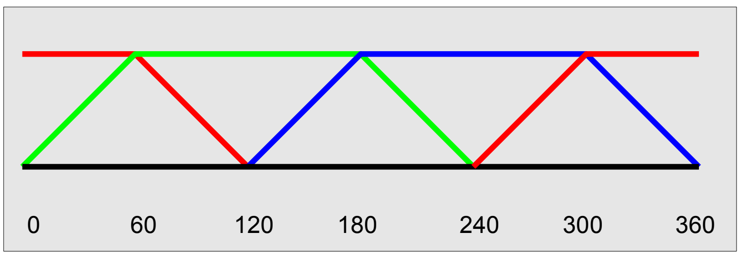


Linear

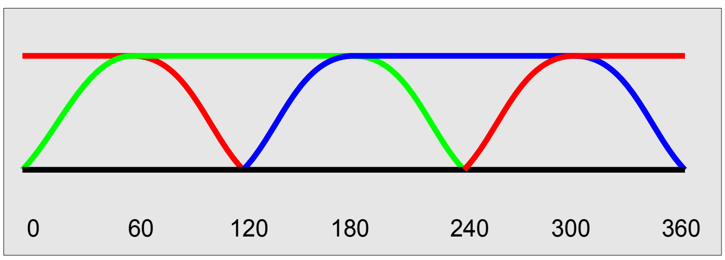


Quadratic

The first is generated using the following distribution primary of colours

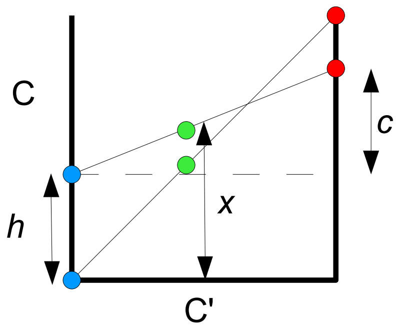


while the second is produced by this algorithm:



Systems which use **hue** as one of their components are called 'cylindrical' or 'conical' systems because one of their components is an angle, not a value. In a computer, the angle is still represented as a number between 0 and 255, however.

In our first attempt at construction a cylindrical system we note that all we need is a simple mathematical function which will turn the components C_R', C_G', and C_B' of the base colour into components of the desired colour.



As we have seen, any linear function will do so lets try the simplest of these functions, namely the equation of a straight line:

$$C = c\,C' + h$$

where c is the gradient of the line and h is the intercept on the axis.

We can, however, do a little better than this by using the following equation:

$$C = c\,C' + x - c/2$$

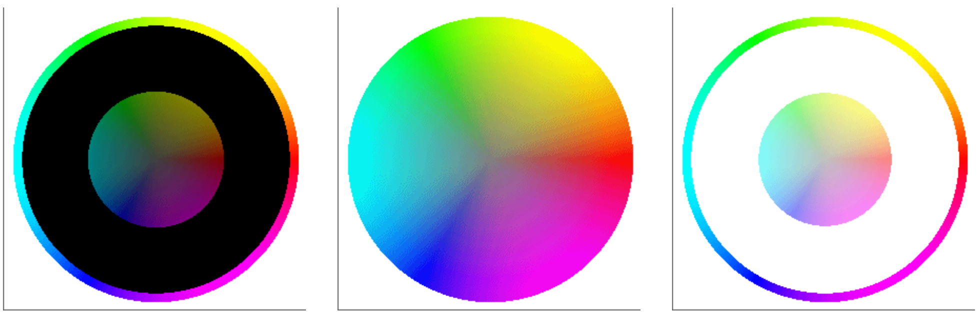
x is the 'median' height of the line rather than the intercept and makes the system more symmetrical with respect to the 'height' variable.

(Note that when using mathematical equations like these, I am assuming that all the parameters lie between 0 and 1. Bear in mind that in a real implementation, they all range from 0 to 255.)

Now what interpretation can we give to c and x? c is the gradient of the line and therefore defines what we might call the *colour contrast*. Certainly when $c = 0$, there is no colour because $C_R = C_G = C_B$. x is a measure of the brightness of the colour.

We now have a choice. We can either plot c radially and vary x. I shall call this system the HC – X system. Alternatively we can plot x radially and alter c. This is the HX – C system. Lets see what they look like.

The HC – X system

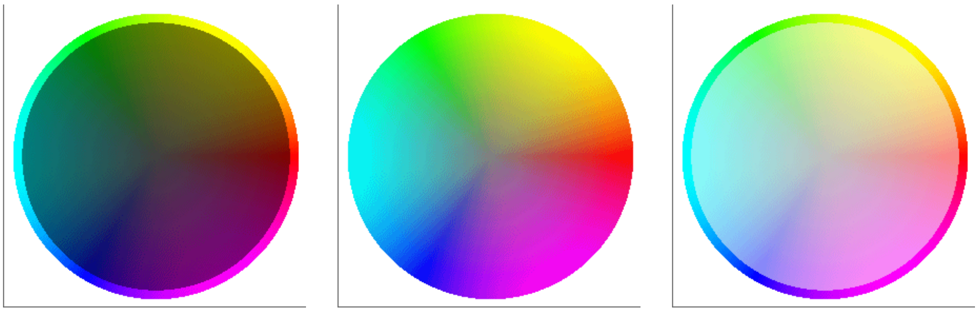


$x = 64$ $x = 128$ $x = 192$

The first thing that you notice about this system is that it has holes in it! This is because, for some values of x some components go either negative or exceed their maximum allowed value. These colours obviously do not exist and have been rendered either black or white. (The ring of base colours round the edge is not part of the system. It is for reference.)

The normalised HC – X system

The nice thing about this system is for every value of x, the wheel displays all the colours that exist with the same median brightness. All that is needed to make it really practical is to expand the small discs to fill the whole circle**.** Then we get a really usable system which looks like this:

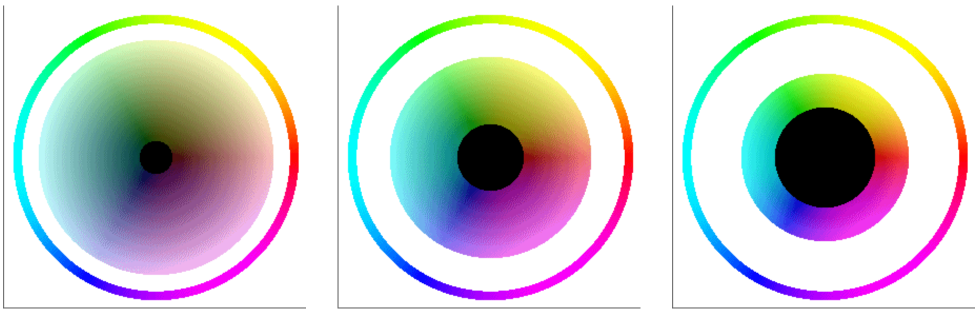


$x = 64$ $x = 128$ $x = 192$

This system is sometimes called the bi-conical system because from $x = 0$ to $x = 128$, the colours expand from the black corner of the colour cube; then in the second half they contract down to pure white.

Before we leave this system, lets have a quick look at its complement, the HX – C system.

The HX – C system



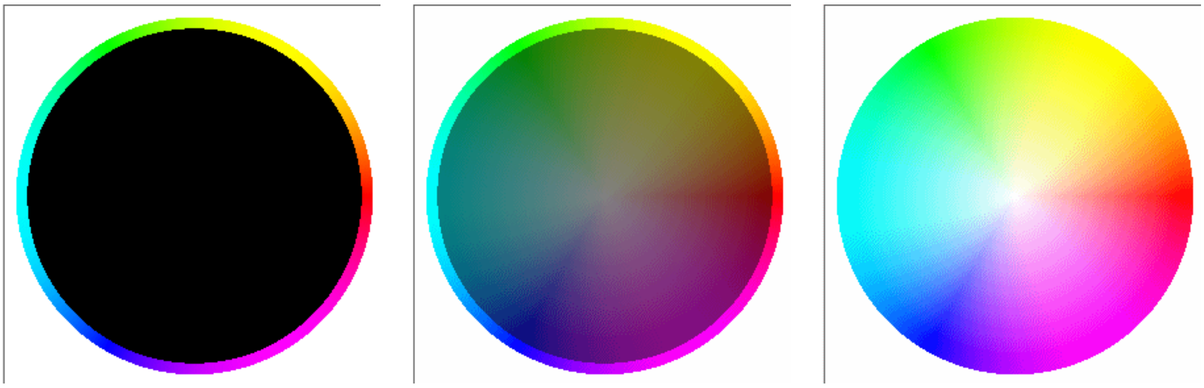
$c = 64$ $c = 128$ $c = 192$

Like the HC – X system, this one has holes and it cannot easily be normalised so we shall reject it.

With the normalised HC – X system being so satisfactory, it is a little difficult to see why the conical system most widely used is the HSV model. Lets see if we can at least understand it.

The HS – V system

In the HS – V system, colours whose **value** is zero (V=0) are black. As the **value** is increased the colours emerge from the shadows as it were. All the base colours have V=255 of course. The following colour wheels show the effect of increasing the **value**. (Incidentally, the **saturation** component (S) increases from 0 in the middle to 255 at the edge.)

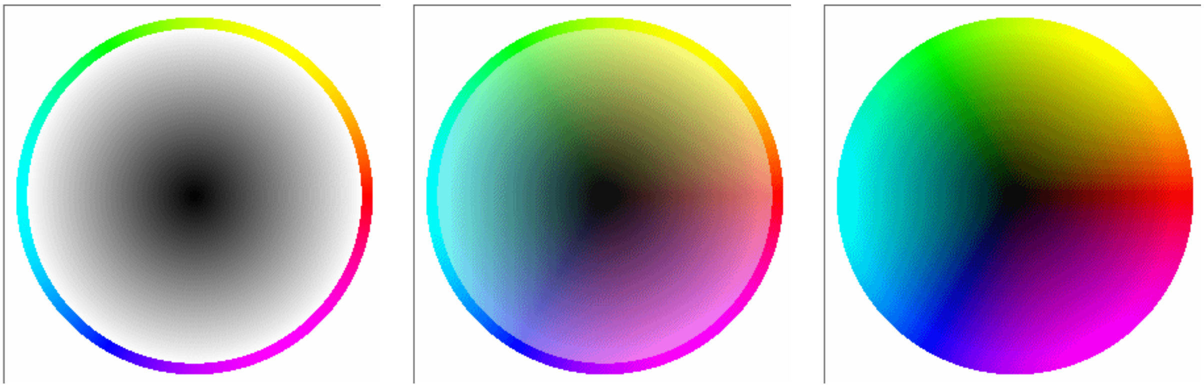


$V = 0$ $V = 128$ $V = 255$

The system is fine for choosing dark colours but all the interesting pastel shades are clustered round the centre of the white spot in the third disc are are consequently rather difficult to find accurately.

The HV – S system

In the HV – S system, colours whose **saturation** is zero (S=0) are shades of grey. As the **saturation** is increased, the colours become more colourful. All the base colours have S=255. The following colour wheels show the effect of increasing the **saturation**.



$S = 0$ $S = 128$ $S = 255$

Where are all those pastel shades now? They seem to have disappeared completely! They are there, of course. Every one of these systems includes every possible colour. They will be found round the edge on the S = 40 disc.

Converting from HSV to RGB

In order to understand the system completely, we need to know how to convert HSV values into RGB values. The first step is to calculate the base colour from the hue. The general principle for doing this has already been outlined and as has been stated already, the base colour always has one component of 255, and one which is zero. To make the formulae simpler to understand, we shall use the symbol C to stand for any component (normalised so that C has the range 0 – 1).

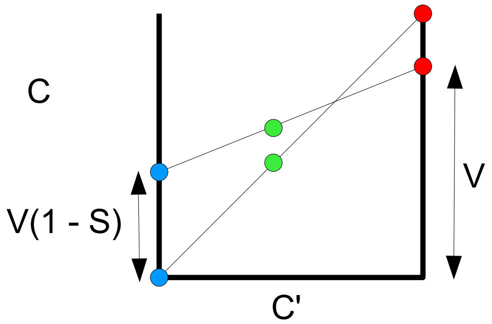
To calculate the actual value C of any component whose base colour value is C', you apply the following simple formula:

$$\mathrm{C} = V(1 - S(1 - \mathrm{C}'))$$

or, if you prefer:

$$\mathrm{C} = SV\mathrm{C}' + V(1 - S)$$

To make this a little easier to understand, here is a diagram of the transformation:



The example shows a base colour of orange. Its red component, $C'_R = 1$ and this has been reduced to C = V. At the same time, the blue component $C'_B = 0$ has been increased to $C = V(1 – S)$. The green component has been moved to lie on the new line as well. Now you can see why when $V = 0$, the colour is black regardless of the hue. You can also see why, when $S = 1$, both ends of the line are at the same height – V, so the colour is grey.

Converting from RGB to HSV

First you must identify which component is the largest and which is the smallest. It is these which determine S and V. From the above diagram you can easily see that the **value** V is simply the largest component C_{max} and that $C_{min} = V(1 – S)$. It is easy to show that $S = (C_{max} – C_{min}) / C_{max}$.

To work out the **hue** angle it is not actually necessary to work out the middle component of the base colour as, as has been noted in the previous section, the middle component stays in the same place with respect to the other two components. All you need is the ratio $(C_{mid} – C_{min}) / (C_{max} – C_{min})$.

Problems with the HSV model

One very important feature of the HSV system is that it has no holes. More precisely, all values of V, S and C' from 0 to 1 give allowed values of C. It is this feature, together with the simplicity of its transformation which is often stated as being the reason why the HSV model has become the standard one. As we have seen, and shall see, it is by no means the only system with these features and it does have its peculiarities.

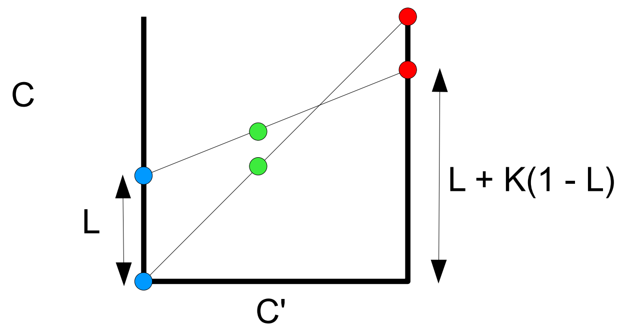
The usual way of presenting the HSV model is to use V as the independently controllable variable. This results in colour wheels like the ones shown under the heading **The HS – V system** which range from completely black to a base colour wheel with white at the centre. Quite apart from the confusion which surrounds the meanings of the words 'saturation' and 'value' (The former is used with different meanings in several systems while the latter doesn't really have any meaning at all), the wheels with V<128 are all but useless because one rarely need lots of subtle shades of near black. On the other hand, the extremely useful pastel shades which cluster round the middle when V is large are difficult to see clearly. It is for this reason that I have devised a different transformation which, I think, results in a much better distribution of colours inside the colour wheel.

The HKL system

The system is called the Hue / Kappa / Lambda system or HKL for short. H (hue) is defined in exactly the same way as in the HSV system but kappa (K) and lambda (L) have new interpretations which I describe below. The transformation formula is:

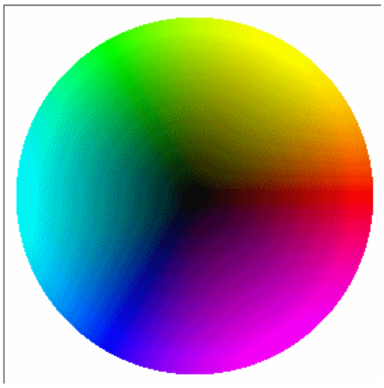
$$C = L + KC'(1 – L)$$

which results in the following graph:

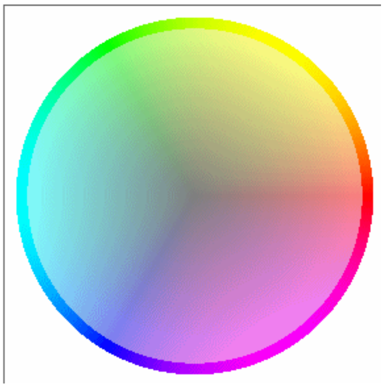


Lets have a look at the colour wheels which we can generate using this system. First we shall explore the effect of changing *L* (lambda):

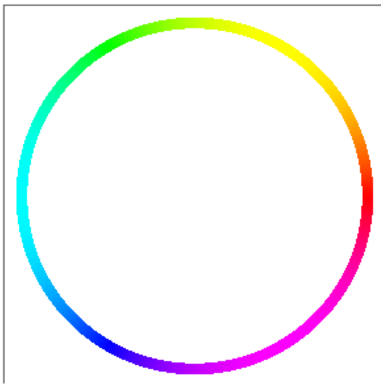
The HK – L system



$L = 0$



$L = 128$



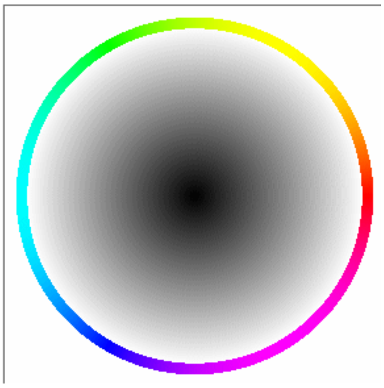
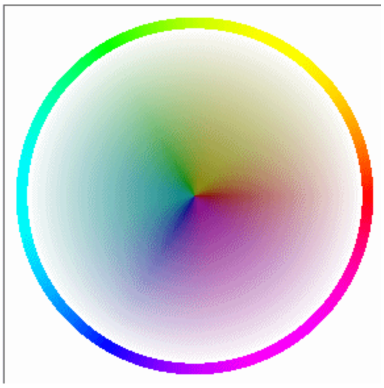
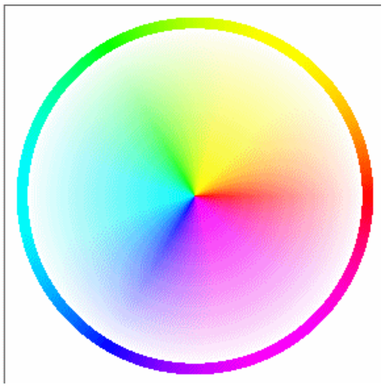
$L = 255$

It will be noticed that **lambda** (L) is similar in a way to **value** (V) in the HSV system in that it alters the 'brightness' of the colour – but instead of starting in the shadows and increasing to full intensity, here the colours start at full intensity and then get lighter ending in pure white. Unlike **value**, however, useful colours are obtained with the full range of **lambda** because, as has already been pointed out, pastel shades are far more useful than dark colours. I have called this component 'lambda' precisely in order to get away from pre-conceived notions attached to words like 'saturation' and 'value' – but if you want an English word to remind you what altering **lambda** does, I suggest thinking of the word 'lightness'.

Try putting $L = 230$ for a lovely disc of pastel colours!

As with all the other colour wheels, the third component is plotted radially. This component is **kappa** (K) and as you can see, at the centre where K=0 we get the greyscale colours with the most colourful colours on the periphery where K is maximum. For completeness, here are three colour wheels showing the effects of changing K on its own.

The HL – K system

$K = 0$ $K = 128$ $K = 255$

Unlike the colour wheels generated by the HK – L system, I do not think these have a great deal to recommend them but they are interesting nonetheless. They clearly show that **kappa** controls what we might call the 'colourfulness' of the colour and in that respect acts a bit like **saturation** in the HSV system.

The relation between the HSV and HKL systems and the colour cube

Both the HSV and the HKL systems are *conical* systems. The former goes from black to full colour while the latter goes from full colour to white. This makes it obvious the two systems are in a fundamental way complementary with white playing the same role in the HLK system as black does in the HSV system. We can see why this is by relating the respective transformation equations to what goes on inside the colour cube.

The HSV transformation is as follows:

$$C = V(1 - S(1 - C'))$$

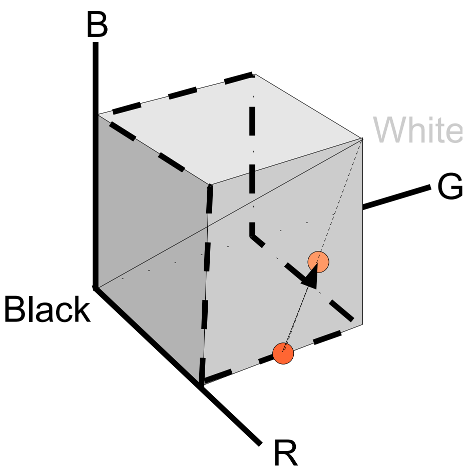
This can be split into two distinct processes:

$$C'' = 1 - S(1 - C')$$

and

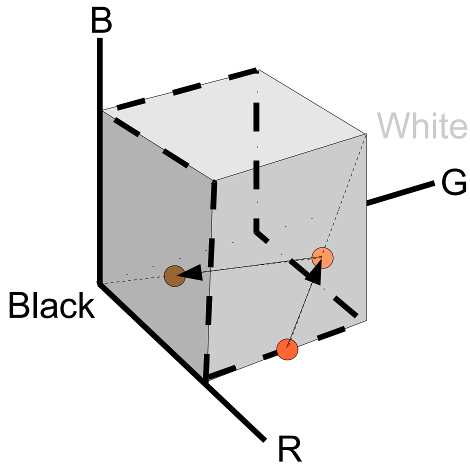
$$C = VC''$$

You will recall that C' represents a component of the **base colour** which lies on the wiggly line joining the principal colours. What the first process does is to move this base colour a certain (proportional) distance towards the white apex of the cube like this:



Here we see how an orange base colour has been moved towards a whiter (less saturated).

The second step multiplies all three components by V. this simply moves the colour a certain distance towards the black apex. This turns the colour brown – ie by reducing its **value**.



The HKL transformation can likewise be split into two parts:

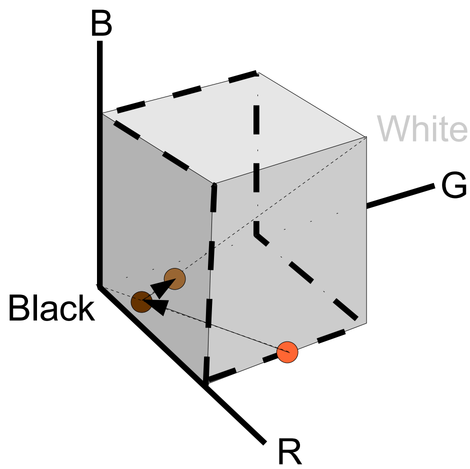
$$C'' = K\,C'$$

and

$$C = L + C''(1 - L)$$

The first step moves the base colour towards black by multiplying by K

and the second step moves it towards white. The following diagram shows how the same colour orange is transformed into the same colour brown by a different route.



This interpretation suggests a third system in which black and white play the same roles and in which the two independent parameters (which I shall call P and Q) pull the colour equally in two directions – P towards black and Q towards white.

The HPQ system

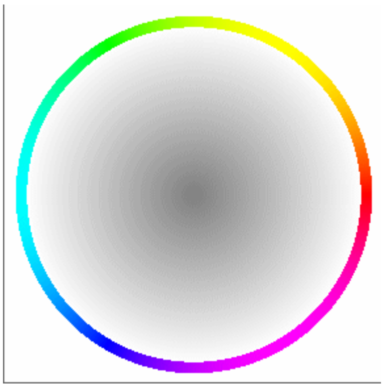
The transformation for this system is

$$C = PQC' + (1 + P)\,(1 - Q)/2$$

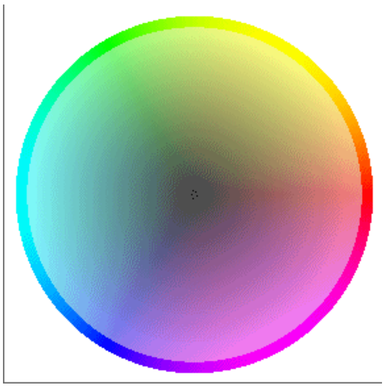
You will notice that, apart from the minus sign, P and Q are symmetrical. You may like to verify that if either P or Q are zero, C is independent of C'; resulting in a greyscale of colours. Also when $P = 1$ and $Q \Rightarrow 0$, C => 1 (ie white) while if $Q = 1$ and $P \Rightarrow 0$, C => 0 (ie black).

This is what the two complementary systems look like:

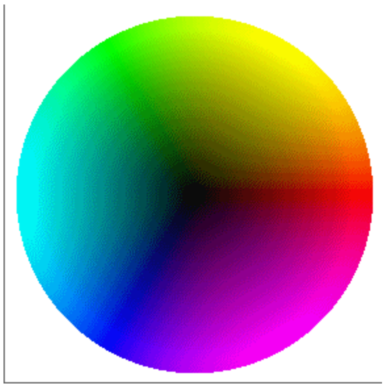
The HP – Q system



Q = 0

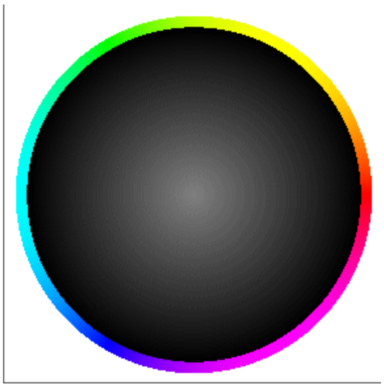


Q = 128

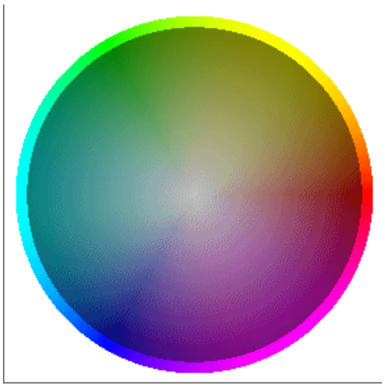


Q = 192

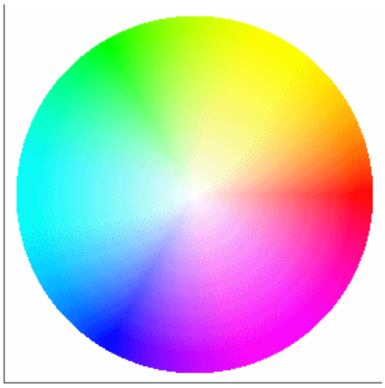
The HQ – P system



P = 0



P = 128



P = 192

Conclusion

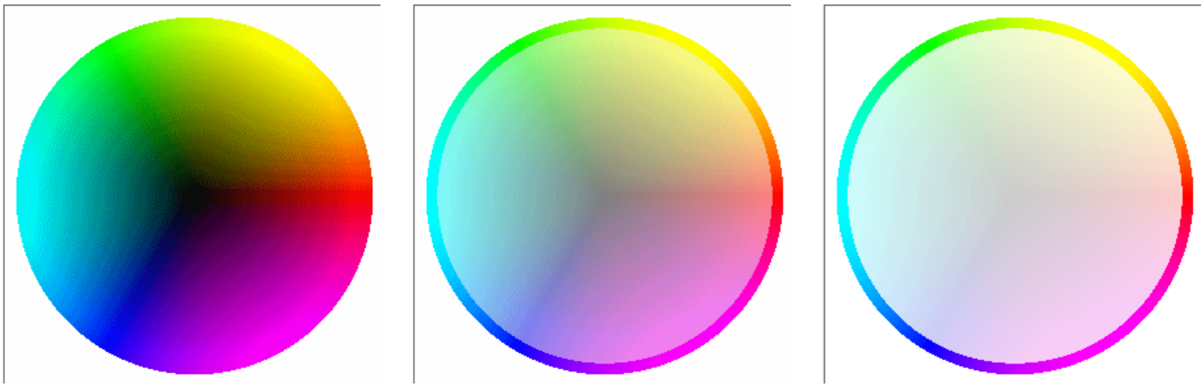
So which is the 'best' system?

I don't think that there is any doubt that any system based on hues is better than any RGB system but it has to be said that of the four alternative cylindrical systems discussed, there is very little to choose between them. Here is a table of what I perceive to be their strengths and weaknesses:

HC - X^*	An excellent system with meaningful parameters. (* = normalised)
HX - C	Useless.
HS - V	Too much emphasis on dark colours.
HV - S	A useful system but there is much confusion as to the meaning of 'saturation' and 'value'.
HK - L	An excellent system with good emphasis on pastel colours.
HL - K	Useless.
HP - Q	Mathematically interesting but not very useful.
HQ - P	Mathematically interesting but not very useful. Too much emphasis on dark colours.

For my money, the quadratic HK – L system is by far and away the best. Just to rub it in – here it is again:

The quadratic HK – L system



$L = 0$ $L = 128$ $L = 192$

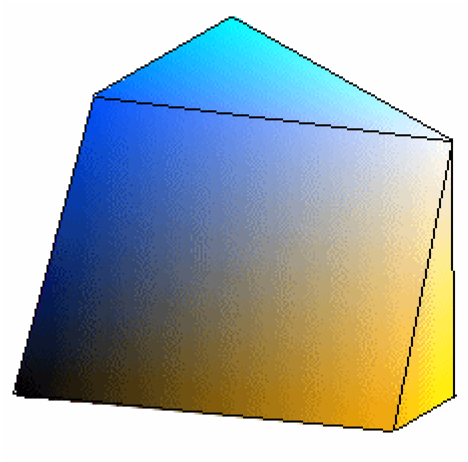
Can you find olive drab pale lilac now?

'Cheese slice' systems

All the conical systems discussed have one feature which is both an advantage and a drawback. Every disc shows every possible hue.

What if you want to choose from all the possible variations of a single hue? Indeed, where do all the possible variations on a single hue lie in the colour cube?

To answer this question we need to remember two things: 1) linear transformations do not change the hue; and 2) the grayscale line from the origin to the opposite corner is common to all hues. What this means is that any planar slice through the cube which includes the long diagonal will include all the possible variations of a single hue (and its complement). Here is an example:

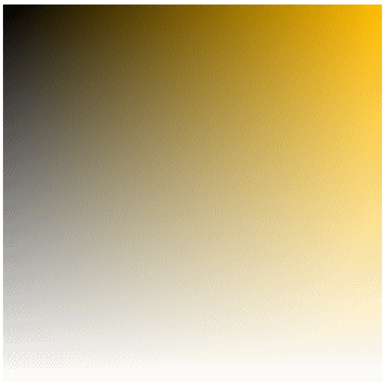


Note how every slice is a rhombus with black and white at opposite ends and two complementary base colours at the other two corners. This rhombus contains all the possible shades of orange and its ultramarine complement.

For certain purposes, the presence of two different colours in the same diagram is a little distracting. We can produce shades of a single colour quite easily by plotting the second and third parameters of any of the conical systems on a cartesian grid and use our slider to select the hue.

The following diagrams show you the effect you get using an orange colour as the basic hue:

SV – H

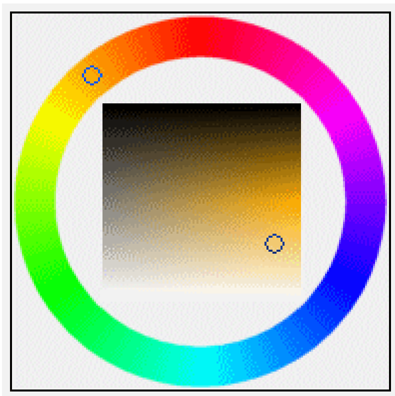


KL – H

PQ – H

As we have noted before, the HSV and the HKL systems are complementary – the former emerging from the shadows from left to right, and the latter rising up from whiteness. In this mode, I think the HPQ system has the advantage because every single point on the square represents a different colour – even along the edges.

Paint Shop Pro (which I use extensively) uses a similar scheme but I cannot see the reason for putting the base colour half way up the right hand side. Nor can I see the advantage of having a large percentage of the square in shades of (nearly) grey.



Overall conclusions

My ideal colour picker would have two modes. 1) a colour wheel mode using the HK – L system and 2) a 'Cheese slice' mode using the PQ – H system.