COLOUR GROUP SELECTION FOR COMPUTER INTERFACES

Paul Lyons[a], Giovanni Moretti[a] and Mark Wilson[b]

[a]Institute of Information Sciences and Technology. Massey University,
Private Bag 11-222, Palmerston North, New Zealand

[b]Harvest Electronics USA Incorporated
Suite 118, 120 Interstate North Parkway East, Atlanta, GA 30339

ABSTRACT

We describe a low-impact method for colouring interfaces harmoniously. The method uses a model that characterises the overall image including the need for distinguishability between interface components.

The degree of visual distinction between one component and other components, and its colour strength (which increases with its importance and decreases with its size and longevity), are used in generating a rigid ball-and-stick "colour molecule," which represents the colour relationships between the interface components. The shape of the colour molecule is chosen to conform to standard principles of colour harmony (like colours harmonise, complementary colours harmonise, cycles in the colour space harmonise, and so on). The colour molecule's shape is fixed, but its position and orientation within the perceptually uniform colour solid are not. The end user of the application chooses a new colour scheme for the complete interface by repositioning the molecule within the colour space. The molecule's shape and rigidity, and the space's perceptual uniformity, ensures the distinguishability and colour harmony of the components are maintained.

The system produces a selection of colour schemes which often include subtle "nameless" colours that people rarely choose using conventional colour controls, but which blend smoothly into a harmonious colour scheme. A new set of equally harmonious colour schemes only requires repositioning the colour molecule within the space.

Keywords: Interface colour, perceptual uniformity, colour molecule, holistic computing, colour selection

1. INTRODUCTION

In choosing colours for an interface, we try to satisfy both aesthetic and pragmatic objectives: the colours chosen should ensure that some pairs of items on the screen can be clearly distinguished; the interface should produce an overall feeling of colour harmony; and the colours of individual interface components should be pleasant. The first two of these objectives concern the visual relationship between multiple screen objects. They are more important than the third, which concerns the appearance of isolated interface components. However, conventional colour pickers only support the third objective. Users are left to their own devices to achieve an interface with an overall colour balance, and clearly distinguishable components. This complicates the design of interfaces with multiple coloured components. We hypothesise:

- that many of the aesthetic and pragmatic criteria characterising commercially important images, such as computer interfaces, can be represented by a mathematical formalism (following Munsell, cited in Birren[1]))
- that this formalism can be conceptualised as an abstract colour scheme
- that the abstract colour scheme can be represented by a "colour molecule", a rigid shape in a 3D colour-space
- that the abstract colour scheme can be converted to a specific colour scheme which conforms to the aforementioned aesthetic and pragmatic criteria by locating the colour molecule within the colour space.

Correspondence: P. Lyons. Other author information: PL: Ph: (+646) 3505799, Fax: (+646) 3502259, email: P.Lyons@massey.ac.nz;
GM: Ph: (+646) 3505799, Fax: (+646) 3502259, email: G.Moretti@massey.ac.nz;
MW: Ph: 770 541 4545, email: MarkWilson@harvest.com

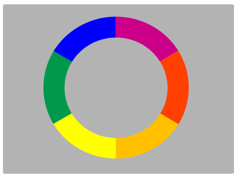

Figure 1. A colour wheel

This paper contains a preliminary description of the Colour Harmoniser (a software tool for assisting application developers to colour their application interfaces by applying these concepts) and a description of some results obtained with a pilot version of the software. The Colour Harmoniser divides colour selection for an interface into two phases. First, colour relationships which are to be maintained between the interface's visual components and are defined and used to construct a rigid *colour molecule* with "interatomic distances" corresponding to the colour distinctions that need to be maintained between the interface components, and a shape that corresponds to one of the common rules for colour harmony.

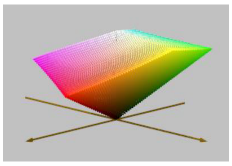

Figure 2. The L*u*v* colour space

Secondly, the user fixes the position and orientation of the model in the colour space by direct manipulation, and thereby determines the actual colours of the interface components. These colours will automatically be harmonious and will maintain important characteristics of the interface such as distinguishability between components, because the method used to construct the molecule integrates the standard generic heuristics for achieving harmonious colour schemes and the specific characteristics of the target interface. The user may reposition the molecule, and thereby select a new set of harmonious colours at any time, without damaging the distinctions.

Three-dimensional, perceptually uniform spaces[2] are at the centre of this approach. These allow us to represent far more sophisticated relationships between coloured items than colour wheels (Fig. 1), which only represent a few colours, and are imprecise about the relationships between colours they do represent. If the space is also perceptually uniform, then distances in the space correspond to perceptual differences between pairs of colours. Commonly used colour spaces, such as RGB, HSV, HSL and CMYK are not perceptually uniform, so the constant colour relationships which are a feature of the Colour Harmoniser are not easily achievable within them. The L*u*v* space[3] (fig. 2) can be mathematically derived from the common RGB space[4] and is usually accepted as giving a reasonable approximation to perceptually uniformity[5].

1.1 Colour selection is important

Colour is ubiquitous. We recognise its strong subliminal effects in many areas. For example[6]:

- Advertisers use various colour combinations to achieve their ends. Bright, highly saturated primary colours convey a sense of urgency and fun, and carefully selected tertiary colours convey a sense of sophistication.
- Political parties - indeed whole political ideologies - are summarised by reference to a single colour ("The Greens," and that clarion call from the 1950s: "Better dead than red")
- Colour-coding is used by many graphic artists to make components of a diagram more immediately recognisable than if they were identified solely by shape or by name.
- In the workplace, colours are found to have significant effects on productivity. Greens are often chosen for their restful effect, but large areas of poorly chosen greens can induce feelings of nausea, and reduce productivity. Inappropriately chosen areas of strong colours can cause eye fatigue.

In general, interfaces with garish colours are undesirable; users should be able to concentrate on the task at hand, without being conscious of the appearance of the interface. That is, the interface should be perceptually visible, but cognitively invisible.

1.2 Colour Selection has become even more important

During the twentieth century, the problem of colour selection has become more important, as opportunities for colouring our environment have increased manyfold.

In earlier times, there just weren't as many colours available as there are now. Nature certainly provided a wide range of beautiful, highly saturated dyes and pigments. However, in association with mordants (the chemicals used to fix colouring agents to yarn), and under exposure to light and atmospheric oxygen, most become unstable or lose their saturation. Furthermore, many of the remaining strong colours (like ultramarine - finely ground lapis lazuli - and Indian yellow - made from the urine of cows fed on mangoes) were rare and expensive. Consequently, only a few bright colours (and dull colours too, for that matter) were used, until nineteenth century organic chemists began to dramatically extend the palette of available and affordable colours. Now, with more money to spend, and a wider range of colours to choose from, we no longer think about colouring as a fortuitous by-product of protection against the elements, or a luxury for special occasions; in fact, we are prepared to spend money to change the appearance of our entire surroundings. In earlier times, we were surrounded by stone, wood, brick and wool, with their natural, often inherently harmonious, colours. Now we can choose highly saturated colours for our plastic and painted world, often in isolation from the environment in which they will be used.

1.3 Colour selection is difficult

Colour space is large. Even acknowledged masters of the area have trouble exploring it. Josiah Wedgewood, for example, performed over 10,000 experiments to find the colour - called, simply, pale blue - which would best contrast with the white of the applied decoration on his jasper ware.[7] Furthermore, most people aren't familiar with three dimensional colour spaces. They don't know how to locate points or measure distances in such a space. They may recognise that simplifications (such as "a colour goes with darker or lighter versions of itself") exist, but lack tools to quantify the colour differences or map the simplification onto a colour space.

Designing a colour scheme in such a state of ignorance is like trying to produce plans for a building without knowing anything about geometry or mensuration. Geometers use set squares, rulers, protractors, and so on for specifying and measuring important relations in Euclidean space, which they can then map onto physical space. However, the physical universe is a much closer analogue of Euclidean space than of the complex perceptual phenomenon we call colour space and, although analogous tools *can* be constructed for manipulating relationships in colour space, they are difficult to use. This is not because colour spaces have a complex geometry - they allow us to represent colour relationships in conventional Cartesian or polar coordinates - but because they are irreducibly three-dimensional. That is, every point in the 3D space is associated with a unique colour, so that it is difficult to produce a meaningful projection of the 3D space onto conventional 2D media.

1.4 Colour interactions are important

Colours never occur in isolation. Our perception of a colour is invariably influenced by simultaneous contrast, a phenomenon exhaustively documented by M.E. Chevreul.[8] Simultaneous contrast is not caused by retinal fatigue (which causes after-images[9]) but by the way our perceptual system has evolved to exaggerate differences between adjacent parts of an image. An orange object adjacent to red will appear more yellow, and adjacent to yellow, will appear more red, than if viewed against a neutral background. A brown surrounded by bright yellow will appear duller than if surrounded by grey. Chevreul's nineteenth-century experiments with yarns in the Gobelin tapestry works showed that apparent colour differences in patterns of carpet with identically-coloured patterns but differently-coloured backgrounds were caused, not by poor quality control in the dyeing of the yarn, but by simultaneous contrast between the pattern colours and the different background colours of the carpets.

The simultaneous contrast phenomenon provides objective verification of the idea that colours interact with each other. There are other interactions which are less objectively verifiable, but even more important in choosing colours. Many have been codified into colour selection heuristics. For example, adjacent or diametrically opposed hues (blue and orange, or red and green) on the colour wheel harmonise, whereas hues that are a medium distance from each other (such as green and orange) do not. Such heuristics seem vague but are widely recognised as ways of achieving harmonious colour combinations.[1, 7, 8, 9, 10, 11, 12, 19, 21]

Interior decorators, fabric designers and architects must take these colour interactions into account if their designs are to be successful. Colours are invariably viewed in the context of other colours, and achieving a pleasant sense of overall interaction is more important than the attractiveness of the individual colours.

Computer interfaces share many of these characteristics. Interface features differ in size and importance; they often have to be colour-coded for differentiation. Some screen objects overlap, and need to have clearly distinguishable colours; others can be guaranteed never to overlap, so their colours can be similar without confusion. In addition, as interfaces are often used for long periods, poorly designed colour schemes will lead to user fatigue.

Software developers are rarely skilled in considering all these factors when choosing a set of colours for an application. They are more likely to choose colours for individual components on the basis of their individual attractiveness.

1.5 Computer programmers are not colour designers

Not only are most system designers total amateurs in designing for colour, but the history of the introduction of colour technology into computing has virtually ensured that they will follow poor colour practices.

Colour displays capable of displaying millions of colours have been widely available since the 1950s. However the cost of memory restricted the number of colours available on early EGA and VGA video cards. This meant that only 16 garish colours were used in early colour-capable PC applications. Although this 16-colour restriction is now long gone, many influential PC applications were developed during this time, and today's developers, even if they weren't active during this era, grew up using applications that were developed at that time, and have essentially been trained to expect computer applications to use the brightest, most highly saturated colours available. Worse, just about the time when 24-bit colour became widely available, and computer applications started to emerge from the clumsiness imposed by restricted colour sets, the World Wide Web has standardised on a 216-colour palette[13], so that web designers' colour choices are being artificially restricted.

To summarise, the ideas used in the Colour Harmoniser have been developed because:

- colour selection is important for computer applications
- colour selection is difficult
- colour selection for interfaces concerns colour interaction more than individual colours
- computer application developers are not trained in colour choice or interaction
- computer applications developers have been inured to garish colours

2. FOUNDATIONS OF THE CURRENT PROJECT

Early in the twentieth century, Albert H. Munsell, an American painter and academic, developed a colour classification system based on an empirical arrangement of colours into a three dimensional space[1]. He was by no means the first to recognise that colour has three dimensions. Goethe, to pick but one famous example, had anticipated him[5], but it was Munsell's numeric colour-"naming" system that was adopted by many colour classification bodies, including the CIE, leading eventually to perceptually uniform colour spaces such as the CIE's L*u*v* space.[3]

Munsell also developed techniques, based on the perceptual uniformity of his colour solid, for colour harmonisation, but the commercial success of his earlier work on colour classification has largely eclipsed these more sophisticated but less profitable developments. From the point of view of its adoption into fine art, Munsell's (1905) colour harmony work, which frequently results in desaturated colour schemes, could hardly have occurred at a less propitious time, as abstraction (including colour abstraction) was then replacing naturalism apparently inexorably. However, Munsell's desaturated harmonious colour schemes were adopted by those with more conservative needs; book designers in particular, who needed to achieve subtle effects with colour balance and harmony, adopted his methods for combining colours in graphic design.[1]

Today's software developers are just such another group. Generally, they have no desire to attain a deep understanding of colour, but have to produce applications that are pleasant to look at and easy to work with. There can be quite complex constraints on the colours used for interface components if text is to be readable, and other diagrammatic components are to be distinguishable from each other and the background, and the overall effect neither nondescript nor garish. Software developers need a systematic way of determining sets of harmonious colours that will conform to such pragmatic constraints. We will show how the techniques proposed by Munsell can be developed into a software tool for assisting developers to express their personal colour preferences, to fulfill the pragmatic requirements of the application, and at the same time automatically produce colour combinations that conform to common rules regarding colour harmony.

Munsell's techniques are based on the idea of a three-dimensional colour space. Provided the brightness of the illumination is bounded this space is also bounded, so it is convenient to refer to it as a colour solid.

The first of the three independent colour dimensions of this solid is called *hue*. Newton's prism[18] split sunlight into five[1] constituent hues, red, orange, yellow, green, blue, and intermediate hues. Our perceptual system synthesises a *non-spectral* sensation, purple, from the two colours at the ends of the spectrum, red and blue, and thus we perceive a cycle of hues. So

---

[1] Newton had mystical pretensions; he believed that there ought to be seven hues, so he added *indigo* and *violet* (which is *not* the extra-spectral colour we call purple) to the five hues we use today.[22]

hue is a measure of a colour's position on a circle traversing red orange, yellow, green, blue and purple and back to red again as well as many nameless intermediate hues. Although this “$1^{1}/_{2}$D” circle is popularly called the *Colour Circle*, and is simplistically regarded as a complete description of the colour universe, it does not by any means encompass the full range of perceivable colours. In order to find the rest of the colours, we must incorporate two other dimensions.

The second dimension of perceptual colour is *value*, or *lightness*. In a printed or painted image, the perceived lightness of a coloured area is directly (but not linearly) related to the amount of incident light reflected by the image. In a video image, the lightness at a point on the screen is directly related to the total amount of light emitted from that point.

A two-dimensional hue/value model is still an insufficient representation of the colour universe. For example, it does not describe the difference between dull colours like brown and beige and their bright versions, orange and yellow. This requires a third dimension called *chroma* or *saturation.* Turning up the colour on a TV set to change the image from monochrome to a full colour increases the saturation of its colours, ideally without altering their hues or values. The greys in the monochrome image are zero saturation colours, with undefined (or any) hue. Again, it is convenient to visualise this in terms of video technology. A pixel is coloured grey when all three of its dots are turned on equally. When their intensity is zero or maximum, the "grey" is black or white respectively. In general, fully saturated colours involve some combination of one (giving red, green, blue) or two (giving orange, yellow, purple) phosphors, and unsaturated colours involve some combination of three phosphors.

Munsell attributes the idea of a symmetrical (spherical) colour solid with the most-saturated colours on its equator to Runge.[1] However, he discovered empirically that, if the positions in the solid of the colours are to accord with our perceptions, the highest saturation yellows and blues are respectively higher and lower than the equator, and respectively outside and inside the surface of the “ideal” sphere. To allow perceived differences between colours to be measured directly, he devised a complete, perceptually uniform, colour space. Distances in any 3D colour space will provide some measure of colour difference, but only in a perceptually uniform space will distances be proportional to the differences that humans perceive between colours. Munsell's original space has been improved upon by MacAdam[2] and Farnsworth[14], and the CIE’s L*u*v* and L*a*b* colour spaces.[3]

Munsell used his colour classification solid as the basis of his colour harmony system. Like other authors, principally Itten[11], he theorised that an image would look harmonious if it produced an overall impression of being "centred" on grey, preferably middle-value grey. In order to ensure this, he calculated the *colour strength* of a colour region as the product of its area, its lightness and its saturation. Although, strictly speaking, saturation is not bounded, it is convenient to regard both lightness and saturation as running from 1 to 10. A colour scheme based on complementary colours *a* and *b* would be balanced if $cs_a . \sum area_a = cs_b . \sum area_b$, where cs_a (colour strength) is $saturation_a \times lightness_a$. A small area of high colour strength *a* would therefore balance a large area[1] of low colour strength *b*.

2.1 Applying Munsell’s approach in practice

It seems unlikely that Munsell intended painters to use his formula precisely but such a precise formulaic approach provides an ideal basis for a tool for allocating colours to components of computer interfaces, which are not fine art, but need to look harmonious. Before it can be applied, however, there are certain practical problems to be solved.

First, interfaces can generally be expected to contain more than two differently-coloured regions, so the balancing has to take into account a variety of colour strengths. Secondly, Munsell's colour space is not easily mapped to RGB, so the CIE L*u*v* colour space will be used instead. Thirdly, defining the interface characteristics is a prerequisite to using this information as a basis for colouring for the interface. It is important to capture as much information about the interface from the developer, but for the system to be usable, it is important to minimise the effort in acquiring this information. Consider the strict application of Munsell’s approach to the development of a simple two-colour (red-green) complementary colour scheme. The designer would first measure the areas to be coloured. There might be a single, solid region of each colour, or more usually, a number of smaller regions; this is of no concern, it is the total area that counts. (Indeed, in the computing environment, it would be simple to subdivide and intermix the areas repeatedly to produce a uniform "dithered grey" screen made up of appropriate numbers of red and green pixels. Although this might be a little stark as a user interface, it would be consistent with Munsell's approach.)

After measuring the areas, the designer would choose a colour for one of them. This would have to be located on a Munsell colour chart to determine its precise hue, chroma and value, to allow the total colour strength of the regions with that colour to be calculated. In the Munsell colour space, hue is angular (although his units are decimal) so the second colour would have a hue 180^0 away from that of the first colour.

The value and chroma of the second colour would be chosen next, so the *(colour strength x area)* of the first colour equalled the *(colour strength x area)* of the second colour. That is, $(area_{C_1}\ x\ value_{C_1}\ x\ chroma_{C_1}) = (area_{C_2}\ x\ value_{C_2}\ x\ chroma_{C_2})$. The designer would have a certain amount of freedom in choosing the value and chroma of the second colour, increasing one while decreasing the other.

Finally, the Munsell colour samples with the chosen values of hue, value and chroma would be matched in printer's ink or paint, and the artwork produced, or in the case of a screen display, mapped to RGB values. Neither is a straightforward process.

This is clearly not a process likely to appeal to many designers. There's too much drudgery involved, both mathematical and mechanical. However, by giving the user access to Munsell's technique *via* an intuitive interface and hiding the calculation, it is possible to reduce the drudgery in formulating a harmonious colour scheme. Interestingly, although many computer applications provide WYSIWYG colour selection interfaces, they rarely simplify *group* colour selection, and the way they project the three-dimensional space onto two dimensions can make the location of a preconceived colour difficult.

3. THE COLOUR HARMONISER

Let us first consider how a purpose-built computer application could simplify the task of choosing a two-colour colour scheme. The designer still has to specify the total area of each colour (possibly by drawing the image), and select one of the colours. A Graphical User Interface can be used to simplify the latter task. The application can then calculate the (*area x colour)* strength product for the first colour, and consequently for the second colour. It then presents the user with a line of colours, all of which have the necessary hue to counterbalance the first colour, and a *(value x chroma)* product, selected to satisfy the requirement that $(area_{C1}\ x\ value_{C1}\ x\ chroma_{C1}) = (area_{C2}\ x\ value_{C2}\ x\ chroma_{C2})$. The user chooses a particular colour from the line and the computer updates the interface with the chosen colours immediately.

None of this has required any calculation by the user, and the colour-matching stages are eliminated. All colour schemes inherently restrict the colourist's choice, but users may nevertheless feel that their freedom to exercise personal choice has been a little compromised (though the choice of the first colour was unrestricted, and there is at least an element of personal preference in the determination of the second colour).

The project described herein is a generalisation of such a system. It will involve developing the Colour Harmoniser, a piece for designing colour schemes for computer interfaces. There are several reasons for choosing this as the first application area. It is an area where the professionals (applications programmers) are unskilled. Therefore, whereas artists might feel that such a mechanised approach was a slight on their artistic integrity, applications designers seem likely to welcome it as a way of improving the results they can obtain without having to learn a whole lot of colour theory. The environment in which a software package will finally be used – on a computer – means that computing power is available. It makes sense to take advantage of this to aid the user in selecting a colour scheme they (rather than the software developer) feel is most appropriate. Finally, the effect of a colour scheme in a computer interface which may be used eight or more hours a day needs to be harmonious if the interface is not to be visually exhausting. There might also be a call for inharmonious colour schemes; warning messages in nuclear power stations, for example, might be better if they didn't have a soothing appearance, and spooky computer games set in dungeons could benefit from a disturbing colour scheme. Such schemes can probably be produced using rules complementary to those identified by Munsell.

The Colour Harmoniser will, in some respects, be more ambitious than might be inferred from the two-colour example described above. It will allow for colour schemes with more than two colours and as most interfaces have a large number of screen objects it must be able to generate colour schemes that maintain overall harmony between many coloured items.

Secondly, in a computer interface, it will be necessary to ensure that most components are clearly distinguishable from the background; that some are distinguishable from each other; and that some are coloured identically. There may also be some pairs of components that do not interact with each other in terms of distinguishability. For example, if A is completely enclosed by B, its colour need have no interaction beyond general colour harmony with any other object.

Thirdly, the planned software takes account of more parameters than just size in determining the colour strength of a colour region in the image. In a computer interface, some components are more important than others; they therefore "deserve" to be coloured more strongly than less important components. Similarly, components that are only on the screen briefly can be more strongly coloured than those on the screen for long periods. Examples of the two extremes are the background for a window (big, long-lived, and unimportant) and warning messages (small, temporary, and highly important), which should be dull-coloured and vivid respectively.

The project also falls short of the above description in one, hopefully unimportant, respect. Intentionally, it does not take into account the *exact* area of interface components. Most applications are not static. Graphical editors, for example, allow

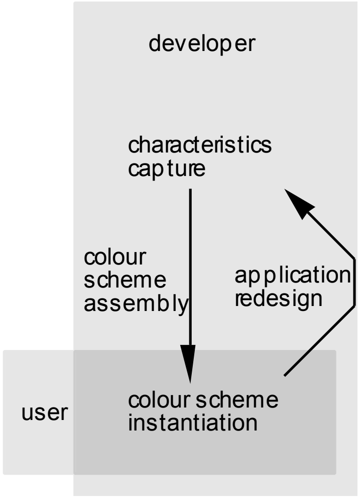

Figure 3. The developer and the end user see the Colour Harmoniser differently

the user to place various interface components on the screen, and computer games change the appearance of the screen more or less unilaterally, though in response to user input. Thus the application designer cannot predict the precise amount of screen real estate occupied by each colour. If Munsell's formula were applied strictly, the colours of screen objects would change when their area changed. This is certainly technically feasible, but would probably also be psychologically disturbing, and the authors feel that a static (though changeable by the user) colour scheme would be preferable. Thus the application developer will have to estimate the area that the various screen components will occupy.

3.1 A brief description

Figure 3 shows a model of the Colour Harmoniser's two-part architecture. Only the application developer will see the first part. It will assemble and abstract colour scheme, called a "colour molecule", using data about the application under development. The second part of the system will be accessible by both the application developer and the day-to-day user. It will allow a colour molecule to be positioned arbitrarily within the colour space so that a specific colour scheme, based on the characteristics entered by the developer, can be instantiated. The interface characteristics specified by the application developer are clearly an important aspect of the system. They include a list of the types of screen object used in the application, and the characteristics of each object type: the total area it occupies, its importance, and its longevity. The application developer will also specify the interaction between each pair of object types. Some pairs will have to be distinguishable. For example, for all "objects$_i$" appearing directly on the background, the interaction (Object$_i$, Background) will be given the interaction "distinct". Some pairs will have a "null" interaction (because they never overlap or abut, so it is not important that their colours are distinct). Some pairs will have to be given "identical" colours such as for example, wires and connection terminals in an electronic circuit drawing package.

Consider the interface for an electronic circuit editor. Figure 4 shows the icon types used in the interface: background, component (AND gate, OR gate, etc), legend (label on a component), wire, terminal (an attachment point for wires).

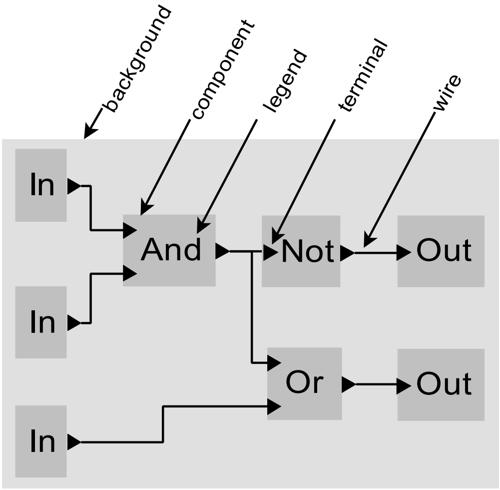

Figure 4. Components to be coloured in a simple electronic circuit diagram

The background is large, unimportant and long-lived. The other objects are all more important and shorter-lived. The background has to remain distinct from all the object types except the legends, which are confined within the components. Components must remain distinct from all the other component types. Wires and terminals should be identically coloured, and distinct from components and the background. These characteristics are summarised in the table shown on the next page in Figure 5 (the *colour strength index* is covered in section 3.5).

3.2 The colour molecule and the abstract colour scheme

The second part of the Colour Harmoniser converts these characteristics of individual interface components and component pairs into an optimal set of colour relationships that must be maintained to ensure a usable interface. Mapping this *abstract colour scheme* onto the colour solid generates a set of points that correspond to a set of interface component colours. However, the mapping takes place in two phases. In the first phase, only the *relative* positions of the colours within the space are defined. This results in a model of the colour scheme which is rigid, but capable of being repositioned. In this respect, it is much like rigid ball-and-stick models of simple molecules that chemistry students build and examine to gain an appreciation of the three-dimensional structure of a chemical species. This mobile model of the colour scheme is therefore called a "colour molecule". The important feature of the colour molecule is that its rigidity maintains the colour relationships between interface components, but its positional mobility

	single-component characteristics			colour strength	component-pair characteristics				
	size (pixels)	importance (0-10)	longevity (0-10)	index (0-1)	legend	wire	terminal	component	background
legend	3	10	7	0.719		null	null	distinct	distinct
wire	10	9	6	0.666	null		null	distinct	distinct
terminal	5	10	8	0.663	null	null		distinct	distinct
Component	25	8	8	0.527	distinct	distinct	distinct		distinct
Background	1000	0	10	0.033	distinct	distinct	distinct	distinct	

Figure 5. Capturing the characteristics of objects making up the target interface

allows the colours themselves to remain unspecified. (This is analogous to building plans, which specify locations in self-relative coordinates; they do not specify absolute geographical locations). Thus the application designer has artistic freedom - within limits - to choose a colour scheme for the interface. The limits are necessary to maintain the necessary colour relationships that ensure the interface is usable.

The second phase of the mapping fixes the molecule in the colour space, thereby associating all of its atoms with specific colours. Since the colour space is three-dimensional, this can be achieved by assigning absolute colour space coordinates to two, or at-most three atoms. Then, the user can experiment with different colour schemes by repositioning and reorienting the colour molecule within the colour space. The rigidity of the colour molecule and the perceptual uniformity of the L*u*v* colour space ensure that the necessary colour differences between the colours will be maintained in any molecular position and orientation. If the space were perceptually non-uniform, two atoms that were clearly distinguishable in one orientation could become indistinguishable in another, even though their Euclidean separation remained constant.

3.3 Constructing the colour molecule

The shape of the colour molecule is obtained by a process analogous to the chemical structure determination technique called molecular packing analysis.[15, 16, 17] In this technique, a model of an n-atom subject molecule is rotated in the regular structure of its crystal, until $\sum_{i=1}^{n}\sum_{j=1}^{m} \text{interaction}_{i,j}$, the total interaction energy between its atoms and the m surrounding atoms, has been minimised. This determines the orientation the molecule will have in crystals of the solid. The thin lines in Figure 6 depict the interactions between one atom in the subject molecule (shown using thick lines) and some of its neighbours within an arbitrary cutoff distance (to simplify the diagram, not all the interactions are shown).

We shall modify Kitaigorodskii's technique slightly to select various configurations of the colour scheme molecules. When dealing with the configuration of chemical molecules, we fix the relative positions of a number of molecules and spin them around till we find the optimum configuration. When dealing with colour molecules, we fix the overall shape of a single molecule so that it corresponds to one of the common colour-selection heuristics (Figure 7), and allow the individual colour atoms to slide along this shape - as though they were beads on a wire frame - to find the optimum configuration.

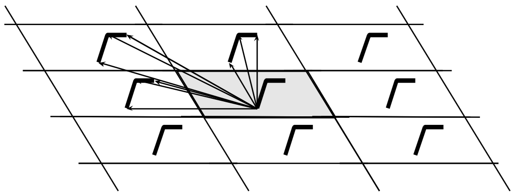

Figure 6. The interaction energy between an atom of the subject molecule and the surrounding atoms is calculated (only some of the interactions are shown)

Harmonious colour schemes can be produced by	Heuristic	The corresponding colour molecule will be
varying the saturation or value of a single hue	*single hue*	a straight line or simple curve between two points in the colour solid with the same hue
combining various saturations of a single hue and grey	*single hue and grey*	a single-hued straight line or simple curve between a point in the colour solid and the grey axis
combining various saturations and values of single hue and black	*single hue and black*	a straight line between a point in the colour solid and black
combining various saturations and values of a single hue and white	*single hue and white*	a straight line between a point in the colour solid and white
combining a various saturations and values of a hue and its complement	*complementary*	a straight line between two points in the colour solid, passing through the grey axis
combining various saturations and values of a hue and two colours on either side of its complement	*split complementary*	a Y-shape from a point A in the colour solid, to two points B and C spaced equal distances away from a point A' directly opposite the first point
combining a set of colours of similar hue	*analogous*	a short horizontal line or arc passing through not more than approximately 25% of the hues
combining colours that extend over the full range of hues	*full-spectrum*	an ellipse

Figure 7. Harmonious colour schemes and the shape of the corresponding colour molecules

3.4. The abstract colour scheme

When the colour atoms are eventually positioned within the space, their positions will define actual colours, but at the optimisation stage, the position and orientation of the molecule within the colour space are unspecified. Thus a colour molecule might have the shape of an equilateral triangle, but would not at this stage be associated with any three specific colours. It would simply specify that the three colours which are eventually chosen by positioning the triangle in the colour space *will be* a certain perceptual distance apart, corresponding to the length of the side of the triangle. Each of the standard rules for deriving a colour scheme can be represented by a simple shape in three-dimensional colour space. The standard rules, and the shapes of the corresponding colour molecules are enumerated in Figure 7.

These heuristics for generating colour schemes are well-known, although Munsell seems to be unique in recognising the full-spectrum scheme as a generalisation of triangular, square, rectangular, and pentagonal schemes described by others, such as Johannes Itten.[11] Munsell also informs us that[1] “any path in the Colour Sphere, and some paths outside it, which are themselves orderly in form and interval will lead through a series of colours which accord, and when used together will render the agreeable sensations which we seek in all colour relations,” although he describes no other such path.

3.5 Using Colour Molecules to Develop a Colour Scheme

In order to determine how the colour atoms are positioned, the user first specifies a type of colour scheme (complementary, split complementary etc) Each type of colour scheme can be represented as a wireframe shape which can be arbitrarily positioned within the colour solid. Atoms corresponding to interface components can be pushed onto the wireframe to a position which satisfies two sets of constraints. First, atoms corresponding to low colour-strength components are allocated positions on the wireframe that are closest to the centre of the 3-D colour solid - in approximately linear colour schemes such as complementary and split-complementary these positions are near the centre of the wireframe - and atoms corresponding to high colour-strength components are allocated positions closer to its extremities. Secondly, a repulsive force, R, is applied to each atom-pair, and the configuration with the lowest overall “force” is determined.

In order to produce large colour differences between small interface components, and between interface component that the system designer has designated as “distinct,” the magnitude of the repulsive force R_{ab} between two interface components, a and b, is inversely related to the sizes of a and b, and directly related to the visual distinction required between a and b, and related to the inverse square of the colour difference between a and b.

```
$R_{ab} = ( A * B ) / C^2$
where:
    $A = 1.7 - 1 / ( 1/\log10(S_a) + 1/\log10(S_b) )$
    $S_a = (\mathrm{sqrt}(\mathrm{height}_a^2 + \mathrm{width}_a^2))$
    { height and width are measured in pixels. For object sizes between 1 and 1640 pixels, A varies between 1 - 0.1
    This function is chosen to accentuate the repulsion if either component is small. }

    B = case distinction$_{ab}$ of
        "distinct":  1;
        "null":      0;
        "identical":
        { the last case is handled in a preprocessing phase by replacing component$_a$ and component$_b$ with a
        single pseudo-component ("ab") which is given colour strength = max(colour strength$_a$, colour strength$_b$),
        and the repulsion $R_{ab,c} = \max(R_{a,c}, R_{b,c})$, for all other components, c}

    C = max(colourdifference$_{ab}$, 0.01)/maxcolourdifference$_{\mathrm{wireframe}}$
    { This produces a value for colour difference that is normalised to the size of the wireframe, so that colour
      differences vary from 0.01 for colour atoms at the same location on the wireframe to 2 for colour atoms at
      opposite ends of the wireframe (Figure 8).}
```

If the application developer has specified a null distinction requirement between two interface components, there is no repulsive force between them. Identically coloured interface components will be tied together, but free to move as a group on the wireframe. Finally, colour atoms of two small components that need to be visually distinguishable will strongly repel each other along the wireframe. After the "interaction force" for the whole system has been minimised, these colour atoms will end up large distance apart in the colour space, and their respective interface components will have very different colours.

The Colour Harmoniser will then adjust the positions of the atoms along the wireframe to minimise the total interatomic repulsive force. The configuration that results will be a (rigid) "molecular structure" defining an abstract colour scheme for the interface. The colour scheme is abstract because it defines only colour differences between the interface components, and not their actual colours. The user can reposition the abstract colour scheme within the colour space to produce a wide range of specific colour schemes. The commonly-used RGB space is non-uniform, so there is no guarantee that clearly distinguishable two interface components coloured (R_1, G_1, B_1) and (R_2, G_2, B_2), with a Euclidean separation of E will remain clearly distinguishable if they are recoloured (R_1', G_1', B_1') and (R_2', G_2', B_2'), even though the second pair of points are separated by the same Euclidean distance, E, as the first pair. Hence the Colour Harmoniser uses the perceptually uniform L*u*v* colour space, because in this space, a Euclidean distance must correspond to the same perceived colour difference, no matter where that distance is located in the space.

Let us now deal with the question of determining how strong or dull the colour of an interface component should be, as this determines the component's atom's initial position on the wireframe, before the optimisation step mentioned above. Each component's "colour strength index" (see Figure 5) is the sum of three contributing parameters, S, I, and L, each varying in the range 0 to (approximately) 0.33, to give a total range of 0 to 1. The Euclidean distance from the centre of the wireframe to its outermost extremity is normalised to 1, and the colour atoms are positioned at a location on the wireframe corresponding to their colour strength index.

Component sizes can vary in the range 1 to 1600 pixels (for commonly available current monitors), so we make S = *0.333-0.1* log(size in pixels)*. The user specifies the component's importance and longevity in the range 0 to 10, so I = *importance/30*, and L = *(10-longevity)/30*. A component that is small, important, and short-lived will thus receive a colour strength index close to 1, whereas a component that is large, unimportant and long-lived will receive a colour strength index close to 0.

Items with a low colour strength index will be placed near the centre of the colour molecule wireframe, and the colour solid, and should therefore be dull-coloured, whereas items with a high colour strength will be placed near the edges of the wireframe, and the colour solid, and should therefore be strongly coloured. Consider, for example, a complementary colour scheme, for which the molecule is a straight line passing through the grey axis.

Figure 8 shows that the colour atom for the background - the interface component with the lowest colour strength – can be positioned at a location on the colour line with a colour strength of 0.033 or –0.033, both of which match the colour strength index calculated for the background. Each of the other four interface components can also be positioned at either of two positions on the colour line, so that 32 different colour molecules can be generated for the electronic circuit drawing interface

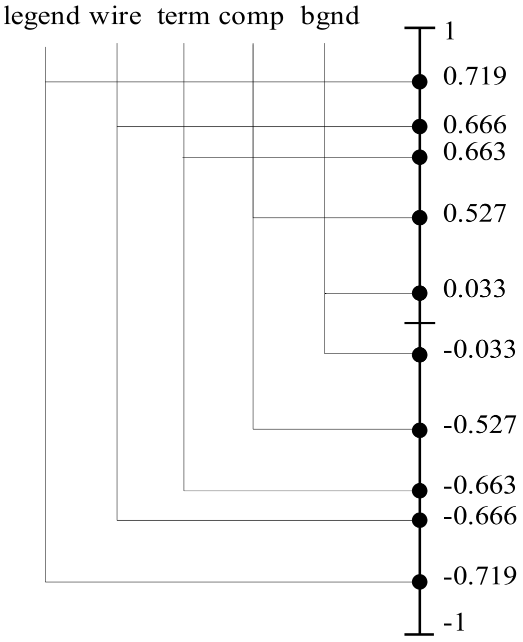

Figure 8. Each colour atom may occupy one of two positions on the colour line

Each of the possible distributions is the starting point for a distinct colour scheme. To find which are closest to the required distinguishability requirements, the "interaction force" of each scheme is evaluated. A small number (6-8) of those with lowest initial interaction force are then optimised further by minimising the force on each colour atom. This corresponds to moving atoms apart that have too high a repulsion force (indicating they wouldn't be distinguishable or would have insufficient colour "impact"). This is done by moving atoms to balance the forces exerted by neighbouring atoms.

The resultant schemes all adhere to the specified distinguishability requirements and the user can select a colour line (as the basis for a range of complementary colours) and click from one to scheme to another to select one that is aesthetically pleasing.

Each "colour molecule" corresponds to an ordering of the interface components along a colour line, optimised to enforce the required distinguishability and "colour impact" requirements laid down in the interface definition table. Therefore, the user, simply by defining the characteristics of the interface objects and choosing two colours to specify a colour line, is presented with a range of possible "abstract colour schemes" overlaid on the colour line they have chosen. Choosing a new end point for the colour line causes the immediate update of the displayed colour scheme.

3.6 The pre-prototype Colour Harmoniser

Parts of the system have been implemented, to test the validity of the concept and the algorithms, and to gain insight into the design of an interface that will hide the model's computational complexity and highlight its perceptual accuracy.

The optimisation algorithm has been tested on the interface components of the electronic drawing package using a hard-coded diagram with a set of user-definable interface parameters (icon size, importance, distinguishability from other icons and so on). The output from the test was a set of eight possible colour schemes. By selecting one of the eight schemes and choosing end-points for the colour line, the user can experiment with different colour schemes, with all colour line and colour molecule changes being instantly reflected in the displayed circuit diagram.

The colour schemes produced are subtle, often unexpected, and mostly harmonious. It is possible to generate garish colour schemes (such as those usually designed by naive software developers) that nevertheless obey the usual precepts of colour harmony by orienting the colour line to combine high values of naturally dark colours (like blue and red), with low values of naturally light colours (like yellow). However, when such a combination is selected, it is only a moment's work, using the interactive direct manipulation interface to alter it to something more pleasing. Because of the ease with which the colour combination can be altered, it seems reasonable to give the user (in addition to the application designer) the ability to manipulate orientation of the colour molecule, and thereby customise the colours for the application while using it. Altering the parameters controlling the position of the colour atoms in the molecule, would remain the responsibility of the application designer.

Naive colourists tend to use hue as the first means of distinguishing objects on the basis of colour, and value as the last; experienced colourists reverse this order. The Colour Harmoniser does not explicitly reinforce the latter order of importance *except* that it colours the picture immediately. It is immediately apparent that a colour molecule oriented horizontally produces poor colouration; correcting this is simply a matter of dragging the molecule to a more vertical orientation.

The Colour Harmoniser enables an application developer to generate a set of abstract colour schemes and select one scheme as a default. The schemes are very easily altered, so there is no reason why the end user should not also be free to select a scheme and rotate these within the colour space and choose whichever suits her or his taste. This gives the end user the freedom to personalise the appearance of the software to their liking, while the application developer can be secure in the knowledge that the pragmatic constraints specified for visual distinction between interface components will not be violated.

4. REFERENCES

1 Birren, F.; *MUNSELL: A Grammar of Color*; Van Nostrand-Reinhold, 1969; pp40-78

2 MacAdam, D.L., *J. Opt. Soc. Am*., **32**, 1942, p247

3 Chamberlin, G.J. and Chamberlin D.G.; *Colour – Its Measurement, Computation and Application*, Heyden, 1980, pp64-68

4 Travis, D.; *Effective Color Displays: Theory and Practice*; Academic Press 1991, pp76-78

5 Norman, R.B.; *Electronic Color*, Van Nostrand Reinhold, 1990, p55

6 Danger, E P; *Using Colour to Sell*, Gower Press, 1968

7 Birren, F.; *Colour;* Mitchell Beazley Arts House, 1980; pp142-145, 217

8 Chevreul, *The Principles of Harmony and Contrast of Colours and their Applications to the Arts,* 1839; quoted by Hilary Page, *The Artist*; **113**, 8, 1998, p23

9 Graves, M.; *Color Fundamentals*, McGraw-Hill, 1952, pp124-128

10 Birren, F.; *The Elements of Color*, Van Nostrand Reinhold, 1970

11 Itten, J., *The Art of Colour*; New York: Van Nostrand Reinhold, 1973

12 Clulow, F.W.; Colour, *Its Principles and Applications*, Fountain Press, London, 1972, 70-71

13 Hess, R.; *The Safety Palette*, Microsoft Developer Network Online, 1996, http://msdn.microsoft.com/workshop/design/color/safety.asp

14 Farnsworth; A temporal factor in colour discrimination*; Visual Problems of Colour*, **II**, Nat. Phys. Lab. Symp. No. 8. HM Stationery Office, 1958 p429, cited in Wyszecki and Stiles[20], 1967

15 Kitaigorodskii, A.I., Organic Crystal Structures; *Adv. in Struct. Res. by Diff. Methods*, **3**, 1970, p173

16 Williams, D.E.; *J Chem. Phys*., **45**, 1966, p3770

17 Williams, D.E.; *J Chem. Phys*., **47**, 1967, p4680

18 Newton, I.; The New Theory about Light and Colours; *Philosophical Transactions of the Royal Society,* **80**, Feb19, 1672, 3075-3087, Quoted in Thayer, H.S., (Ed.) *Newton's Philosophy of Nature*, Hafner, 1953

19 Arnheim, R.; *Art and Visual Perception*, University of California Press, 1974, pp346-364

20 Wyszecki and Stiles, *Color Science*, Wiley 1967

21 von Goethe, J.; *Theory of Colours*, MIT Press, 1997, pp316-370.

22 Leslie, Sir John; *Treatises on various subjects of natural and chemical philosophy*, p59 cited in translator's preface to von Goethe, *Theory of Colours*, MIT Press, 1997, pXXXII

Citation for this paper:

Lyons, P., G. Moretti, and M. Wilson. *Colour Group Selection for Computer Interfaces*. in *Human Vision and Electronic Imaging V*. 2000, ed. B.E. Rogowitz and T.N. Pappas. Vol. 3959. San Jose, USA: SPIE - The International Society for Optical Engineering. 302-313.