A Perceptual Colour Segmentation Algorithm

Christopher G. Healey and James T. Enns

Department of Computer Science
2366 Main Mall
University of British Columbia
Vancouver, British Columbia, V6T 1Z2

e-mail: healey@cs.ubc.ca

Abstract

This paper presents a simple method for segmenting colour regions into categories like red, green, blue, and yellow. We are interested in studying how colour categories influence colour selection during scientific visualization. The ability to name individual colours is also important in other problem domains like real-time displays, user-interface design, and medical imaging systems. Our algorithm uses the Munsell and CIE LUV colour models to automatically segment a colour space like RGB or CIE XYZ into ten colour categories. Users are then asked to name a small number of representative colours from each category. This provides three important results: a measure of the perceptual overlap between neighbouring categories, a measure of a category's strength, and a user-chosen name for each strong category.

We evaluated our technique by segmenting known colour regions from the RGB, HSV, and CIE LUV colour models. The names we obtained were accurate, and the boundaries between different colour categories were well defined. We concluded our investigation by conducting an experiment to obtain user-chosen names and perceptual overlap for ten colour categories along the circumference of a colour wheel in CIE LUV.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces–*ergonomics, screen design (graphics, colour)*; I.3.6 [**Computer Graphics**]: Methodology and Techniques–*ergonomics, interaction techniques*

General Terms: Design, Experimentation

Additional Key Words and Phrases: CIE LUV, CIE XYZ, colour, colour names, intensity, human vision, Munsell, perception, RGB, scientific visualization

1. INTRODUCTION

Many different colour models exist in computer graphics. Each model uses its own 3D coordinate system to identify uniquely individual colours. Some models (*e.g.* CIE XYZ, CIE LUV) are capable of representing all colours from the visible colour domain. Other models (*e.g.* RGB, HSV) are restricted to a subset of this domain. Certain models (*e.g.* CIE LUV, CIE Lab, Munsell) have been designed to try to provide other useful properties like isoluminance and control over perceived colour difference.

The work described in this paper is related to our research in scientific visualization. One of the problems we are studying is how to choose colours to effectively represent multidimensional data. This requires precise control over colour difference. There is evidence that the names which a user attaches to different colours can affect the perceived difference between the colours [Kawai et al., 1995]. If this effect is significant, we will need to consider it during the design of our visualization techniques. This result would also be important for other problem environments which display information using differences in colour (*e.g.* real-time displays, user-interface design, medical imaging systems). We are currently investigating the effects of colour names on colour selection during the design of scientific visualization tools. In order to do this, we need a method for dividing colours into individual categories.

Previous work has described descriptive sets of colour names which can be used to identify accurately colours from the visible colour domain [NBS, 1976; Berk et al., 1982; Tominaga, 1985; Kaufman, 1986]. This work also provides an explanation of how to locate each name within the Munsell colour space. Unfortunately, it is not obvious how to use these results to automatically segment colour models like CIE XYZ or RGB. We wanted an algorithm which was:

- *automatic:* the algorithm should automatically segment a colour region into a set of colour categories; individual colours would then be assigned the name of the category in which they lie;
- *accurate:* the names for each category must be "descriptive", that is, given the set of colour names being used, observers should agree the name chosen by the algorithm represents correctly the colours being named;
- *stable:* the algorithm should build stable, well defined boundaries between neighbouring categories;
- *perceptually controlled:* the algorithm should consider perceived colour differences during naming

Our paper begins with a brief overview of the CIE XYZ, CIE LUV, Munsell, and RGB colour models we used during our colour naming investigation. This is followed by a description of our segmentation technique. We evaluated our algorithm by partitioning known colour regions from the RGB, HSV, and CIE LUV colour models. Results showed that the names were accurate, and the boundaries between colour categories were well defined. Finally, we describe a simple experiment in which observers were asked to name a set of representative colours from ten colour categories along the circumference of a colour wheel in CIE LUV. We show how results from this experiment can be used to measure category overlap and to classify categories as strong or weak. Our results can also be used to identify a user-chosen name for each strong category.

2. COLOUR MODELS

What we commonly call colour is actually our perception of light waves from a thin band of frequencies within the electromagnetic spectrum. This region of visible light ranges from about 4.3×10^{14} hertz to about 7.5×10^{14} hertz. Individual colours are identified by their dominant wavelength λ (which represents hue), excitation purity (which represents saturation), and luminance (which represents intensity). We often refer to colours by their dominant wavelength. Using this notation, colours range from about 400 nanometres ($1\,\mathrm{nm} = 10^{-9}\,\mathrm{m}$) for violet to about 700 nm for red.

Computer scientists use colour models to describe different colours. A colour model is a 3D coordinate system used to identify uniquely colours from a particular colour gamut. Common colour models include RGB (used for colour monitors), CMY (used for colour printers), YIQ (used for colour TV broadcasting), HSV, CIE XYZ, CIE LUV, CIE Lab, and Munsell. Colour models differ both in their coordinate systems, and in the gamut of visible colours they can describe. We used the CIE LUV and Munsell colour models to automatically segment colour regions in RGB, HSV, and CIE LUV.

2.1 CIE XYZ Colour Model

Different-coloured lights can be combined to produce a wide range of colours. One of the most commonly used methods is the combination of red, green, and blue (**R**, **G**, and **B**). The colour matching curves $\overline{r}(\lambda)$, $\overline{g}(\lambda)$, and $\overline{b}(\lambda)$ define the amount of **R**, **G**, and **B** needed to match a colour with a dominant wavelength of λ. An important point to note is that the red curve $\overline{r}(\lambda)$ is negative from 438 nm to 546 nm. Colours from this region of the visible frequency domain cannot be produced through a positive combination of **R**, **G**, and **B**.

In 1931 the Commission Internationale de L'Èclairge (CIE) addressed the problem of negative weights in the RGB colour model. They defined three new primaries called **X**, **Y**, and **Z** to replace **R**, **G**, and **B** during colour matching. Colour matching curves $\overline{x}(\lambda)$, $\overline{y}(\lambda)$ and $\overline{z}(\lambda)$ define the amount of **X**, **Y**, and **Z** needed to match a colour with a dominant wavelength of λ. These curves were designed to be linear combinations of $\overline{r}(\lambda)$, $\overline{g}(\lambda)$, and $\overline{b}(\lambda)$. None of $\overline{x}(\lambda)$, $\overline{y}(\lambda)$ or $\overline{z}(\lambda)$ are negative in the range 380 nm to 780 nm, which means any visible colour can be produced by a positive combination of **X**, **Y**, and **Z**.

Suppose the amount of $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ needed to match some colour $\mathbf{C}$ is defined to be (X, Y, Z); that is, $\mathbf{C} = X\mathbf{X} + Y\mathbf{Y} + Z\mathbf{Z}$. Chromaticity values (x, y, z) are defined by normalizing over $X + Y + Z$:

$$x = \frac{X}{X+Y+Z}\ , \quad y = \frac{Y}{X+Y+Z}\ , \quad z = \frac{Z}{X+Y+Z} \tag{1}$$

The chromaticity values (x, y, z) depend only on the hue and strength of colour $\mathbf{C}$. Luminance information is lost during normalization. Since $x + y + z = 1$, we can recover z from x and y. To obtain the original (X, Y, Z) values, we also need luminance Y. Given (x, y, Y), we can recover the corresponding (X, Y, Z) by:

$$X = \frac{x}{y}Y\ , \quad Y = Y\ , \quad Z = \frac{1-x-y}{y}Y \tag{2}$$

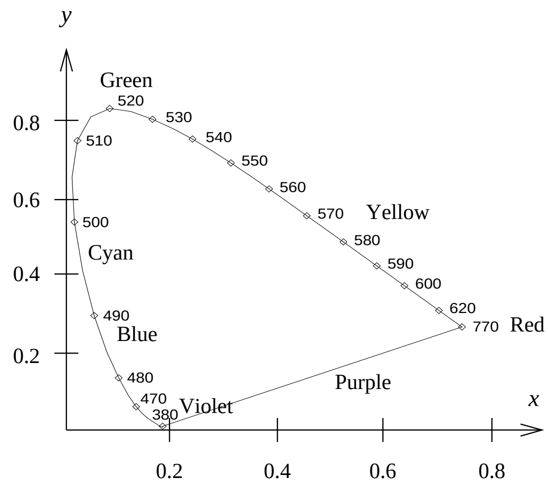


Figure 1: CIE chromaticity diagram, wavelengths (in nanometres) and positions of saturated colours specified on the boundary of the horseshoe

A plot of (x, y) values for all visible colours produces the CIE chromaticity diagram shown in Figure 1. Points on the boundary of the horseshoe represent fully saturated colours. Colours with the same chromaticity but different Y-values project onto the same point in the horseshoe. Since the colour gamut of CIE XYZ is the entire range of visible colours, algorithms are normally provided to convert colours from other models into CIE XYZ values.

2.2 CIE LUV Colour Model

One problem with the CIE XYZ colour model is its lack of perceptual balance. Colours which are the same distance from one another are not necessarily perceptually equidistant. In 1976, the CIE proposed the CIE LUV colour model to address this problem. CIE LUV is a perceptually uniform colour space. This means that distance and difference can be interchanged as required. If colours $\mathbf{A}$ and $\mathbf{B}$ are twice as far apart as colours $\mathbf{C}$ and $\mathbf{D}$, then the perceived difference between $\mathbf{A}$ and $\mathbf{B}$ is roughly twice the perceived difference between $\mathbf{C}$ and $\mathbf{D}$.

The equations for computing CIE LUV assume you have (X, Y, Z) of the colour to convert, and (X_w, Y_w, Z_w) of a standard white. Given these values, the corresponding LUV colour is:

$$\begin{aligned} L^* &= 116(Y/Y_w)^{\frac{1}{3}} - 16, \quad Y/Y_w > 0.01 \\ u^* &= 13L^*(u' - u'_w) \\ v^* &= 13L^*(v' - v'_w) \end{aligned} \tag{3}$$

$$u' = \frac{4X}{X + 15Y + 3Z}, \quad v' = \frac{9Y}{X + 15Y + 3Z} \tag{4}$$

$$u'_w = \frac{4X_w}{X_w + 15Y_w + 3Z_w}, \quad v'_w = \frac{9Y_w}{X_w + 15Y_w + 3Z_w} \tag{5}$$

L^* encodes the luminance or intensity of a given colour, while u' and v' control its chromaticity. CIE LUV provides a metric ΔE^* which can be attached to different colours. ΔE^* measures the distance in CIE LUV space from standard white to a given colour. Colours with the same ΔE^* are perceptually equal.

$$\Delta E^* = \sqrt{(L^{*2} + u^{*2} + v^{*2})} \tag{6}$$

2.3 Munsell Colour Model

The Munsell colour model was initially proposed by Albert H. Munsell in 1898 [Birren, 1969]. It was later revised by the Optical Society of America in 1943 to more closely approximate Munsell's desire for a functional and perceptually balanced colour system. A colour from the Munsell colour model is specified using the three "dimensions" hue, chroma, and value (Figure 2).

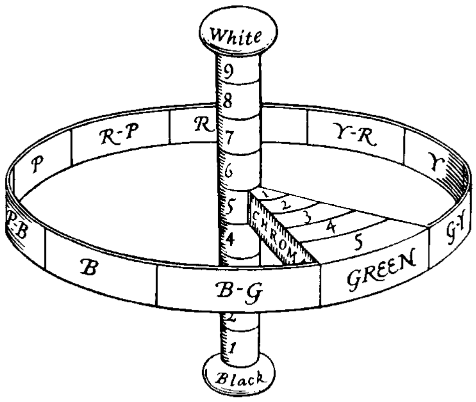


Figure 2: Munsell colour model, showing its three dimensions hue, value, and chroma (from *Munsell: A Grammar of Color*, New York, New York: Van Nostrand Reinhold Company, 1969)

Hue refers to some uniquely identifiable colour. It is represented by a circular band divided into ten sections. Munsell named these sections red, yellow-red, yellow, green-yellow, green, blue-green, blue, purple-blue, purple, and red-purple (or R, YR, Y, GY, G, BG, B, PB, P, and RP for short). Each section can be further divided into ten subsections if finer divisions of hue are needed. A number preceding the hue name is used to define the subsection (*e.g.* 5R or 7BG).

Value refers to a colour's lightness or darkness, and is divided into eleven sections numbered zero through ten. Dark colours have a low value, while lighter colours have a higher value. Value zero represents black, and value ten represents white.

Chroma defines a colour's strength or weakness. Chroma is measured in numbered steps starting at one. Weak colours have low chroma values. Strong colours have high chroma values. Greys are colours with a chroma value of zero. The maximum possible chroma depends on the hue and value being used.

The Munsell model provides a number of useful properties, including perceptual balance and isoluminance. Wyszecki and Stiles provide (x, y, Y) values for a large number Munsell colours [Wyszecki and Stiles, 1982]. In their tables, individual hues are divided into four subsections: 2.5, 5, 7.5, and 10. This provides a total of 40 different hues, plus nine values for each hue and chromas for every hue/value pair.

2.4 RGB Colour Model

The RGB colour model is used by most colour CRT monitors. Different amounts of the monitor's red, green, and blue are added together to produce different colours. The gamut of an RGB monitor is often drawn as a unit cube. The monitor's fully saturated red, yellow, green, cyan, blue, and magenta, along with black and white, are situated at the corners of the cube. The diagonal line from black to white represents shades of grey.

The gamut of an RGB colour cube depends on the chromaticities of the monitor's triads, and the luminance of its maximum-intensity red, green, and blue. Given the triad chromaticities, the monitor's gamut can be drawn as a triangle in the CIE chromaticity horseshoe (Figure 3a). Various texts on colour [Wyszecki and Stiles, 1982; Foley et al., 1990] describe how to build a matrix using these values to convert from monitor RGB to CIE XYZ.

3. SEGMENTING COLOUR REGIONS

Although colour models allow us to describe colours in an unambiguous manner, in practice colours are often identified by name. This technique is not precise, since it relies on a common colour vocabulary and an agreement on what to call different colours. In spite of this, it is useful to be able to name individual colours. Identifying colours by name is a method of communication which everyone understands. Names can also be used to divide a colour gamut into categories of similar colours. That is, given a set of n names $N_1, \ldots, N_n$, we can divide a colour region into n categories $C_1, \ldots, C_n$ such that all the colours in category C_i are best described by the name N_i.

There is evidence that a difference in names increases the perceived difference between pairs of colours [Kawai et al., 1995]. Suppose we choose three colours **A**, **B**, and **C** from CIE LUV such that the distance between **AB** is equal to the distance between **AC**. Moreover, assume that **A** and **B** come from the same named category, but that **C** comes from a different one. In spite of the perceptual balance provided by CIE LUV, viewers might see a larger colour difference between **A** and **C** compared to **A** and **B**. The fact that viewers identify colours **A** and **C** as coming from different named categories increases the perceived difference between them. In order to exploit this property we need a simple method for attaching names to any colour from a given colour region.

The National Bureau of Standards developed the ISCC-NBS colour naming system [NBS, 1976] to try to provide a standard method for choosing colour names. English terms along the three dimensions hue, lightness, and saturation are used to represent different colours. Words from each dimension are combined in various ways to produce 267 different colour names (*e.g.* dark red, strong blue, greenish blue, and so on).

One problem with the ISCC-NBS model is the lack of a systematic syntax; this was addressed during the design of a new Colour-Naming System (CNS) [Berk et al., 1982; Kaufman, 1986]. The CNS was based in part on the ISCC-NBS model. It uses the same three dimensions of hue, lightness, and saturation. However, the rules used to combine words from these dimensions are defined in a formal BNF syntax. Berk compared the RGB, HSV, and CNS colour models by asking observers to use each system to name a set of colours. Observers were most accurate when they used the CNS model. Both Berk and Kaufman describe a method of representing CNS names (as discrete points) within the Munsell colour model.

An extension of CNS called the Colour-Naming Method (CNM) was proposed by Tominaga [Tominaga, 1985]. The CNM uses a systematic syntax similar to the one described in the CNS model. In addition, colour names from the CNM can be mapped to colour ranges in the Munsell colour model. Names in the CNM are specified at one of four accuracy levels called fundamental, gross, medium, or minute classification. Names from a higher accuracy level correspond to smaller colour regions in Munsell. A brief description of the algorithm used to convert from a name to a Munsell colour range is provided in the paper.

In psychology, colour naming experiments are used to divide a colour gamut into named regions. First, the colour names to be used during the experiment are chosen. The colour gamut to be named is divided into representative colours. Observers view these colours one after another and choose from the group of colour names the most appropriate name for each. Experiments of this type are complicated for a number of reasons. The colour gamut has to be divided into a reasonable number of different colours, to ensure that accurate category boundaries are reported. Even a 2D slice through a colour model may need to be divided into hundreds or sometimes thousands of representative colours. A large number of observers are also required, to ensure the choice of names is not biased in some unusual manner. Different observers will disagree on the name for an individual colour, so some method must be devised to choose a single name in these cases. Finally, some skill is required in picking the initial group of colour names. A set of names which does not cover completely the colour gamut will force users to name certain colours in an ad-hoc fashion (*e.g.* asking observers to name blues from the RGB colour cube using only the names red, green, and yellow).

The CNS, CNM, and ISCC-NBS model are all designed to provide a set of names which can be used to accurately identify colours from the visible colour domain. The CNS and CNM position their names within the Munsell colour model. In particular, the CNM can be described as a "renaming" of Munsell patches, since each name from the CNM is mapped to a colour range in Munsell. Perceptually balanced colour models like Munsell and CIE LUV were created using controlled psychological experiments. Rather than building a new colour naming language or performing our own colour naming tests, we decided to use the built-in properties of Munsell and CIE LUV to provide names for individual colours.

3.1 Method

Our technique assumes the target colour to be named is specified using CIE XYZ. Algorithms exist to convert colours from most colour models into an (x, y, Y) value. The target colour is then mapped from CIE XYZ into the Munsell colour space. The Munsell hue dimension uses colour names to identify different hues; this dimension is used to name the target colour.

There are no simple functions to convert a colour from CIE XYZ to Munsell. Some mathematical algorithms have been proposed [Miyahara and Yoshida, 1988]. The CIE suggests using CIE Lab values to obtain Munsell hue, value, and chroma [CIE, 1976] as:

$$\begin{aligned} H &= tan^{-1}(b^*/a^*) \\ V &= L^*/10 \\ C &= ({a^*}^2 + {b^*}^2)^{\frac{1}{2}} \end{aligned} \tag{7}$$

Unfortunately, these methods are complicated and sometimes inaccurate for certain regions of CIE XYZ or Munsell. We decided to use a much simpler method of mapping, namely table lookup. The Munsell values listed in Wyszecki and Stiles [Wyszecki and Stiles, 1982] are specified as (x, y, Y) values. The Munsell patch closest to the target colour is used to represent the target. Given the target colour (x_t, y_t, Y_t) and a Munsell patch (x_M, y_M, Y_M), the distance between the two is simply:

$$d = \sqrt{(x_t - x_M)^2 + (y_t - y_M)^2 + (Y_t - Y_M)^2} \tag{8}$$

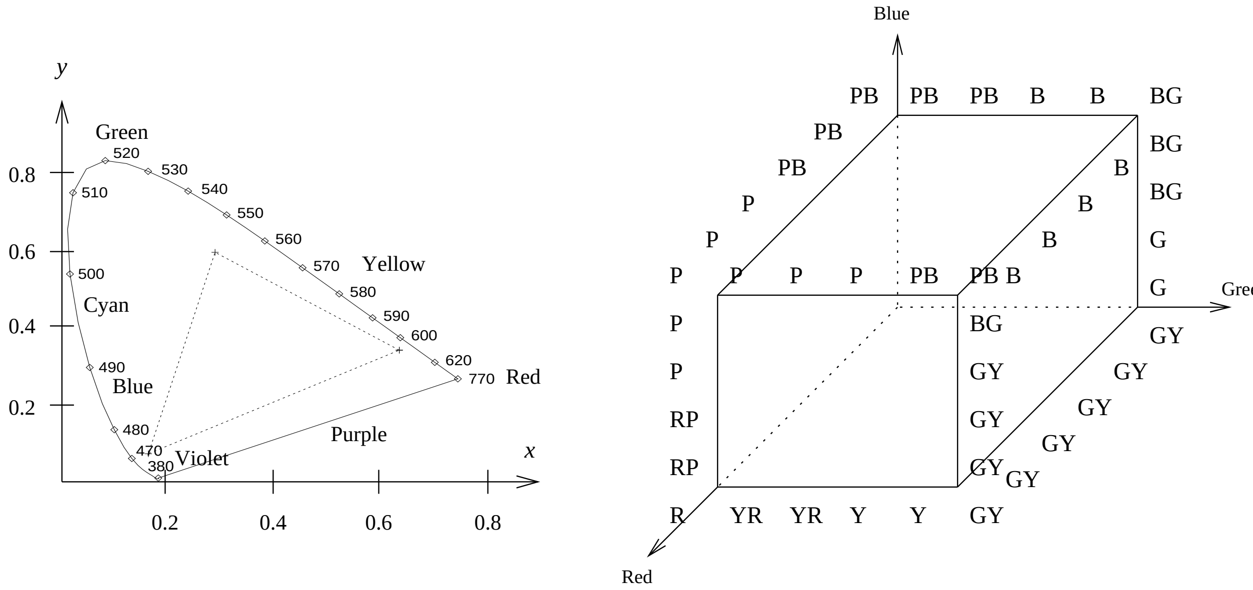
One problem with this technique is the lack of perceptual balance in CIE XYZ. The Munsell patch which gives the smallest d is closest in Euclidean distance, but it may not be closest in perceptual difference. To overcome this, we computed the distance between target and patch in CIE LUV. The Munsell patch colours were converted and stored as (L^*, u^*, v^*) values. Target colours were mapped from CIE XYZ to CIE LUV in a similar manner. Given the target colour (L_t^*, u_t^*, v_t^*) and a Munsell patch (L_M^*, u_M^*, v_M^*), the perceived difference between the two is now:

$$d = \sqrt{(L_t^* - L_M^*)^2 + (u_t^* - u_M^*)^2 + (v_t^* - v_M^*)^2} \tag{9}$$

This allows us to match the target colour to the Munsell patch which is closest in perceptual difference. The hue name of the given patch is then assigned to the target colour.

3.2 Results

When we designed our algorithm, one property we wanted to try to provide was stability, namely that category boundaries were well defined, and that names attached to a smoothly changing set of colours also changed smoothly.



(a) (b)

Figure 3: Silicon Graphics monitor gamut and corresponding RGB colour cube: (a) the the monitor's gamut is defined by the chromaticities of its triads to be a triangle in the CIE chromaticity horseshoe; (b) names for colours along nine of the 12 RGB colour cube edges

Individual Munsell patches are represented as points when they are moved into the CIE LUV coordinate system. Since this is a continuous mapping, the topology of the Munsell colour model is preserved; patches which were adjacent in Munsell are also adjacent (as points) in CIE LUV.

Target colours are associated with Munsell patches using a Euclidean distance metric. Consider the region around a Munsell patch M_i such that all points in the region are closer to M_i than to any other patch M_j. Any target colour which falls within this region is associated with M_i. A 3D Voronoi diagram in CIE LUV of the points representing Munsell patches builds exactly these regions for every M_i [Okabe et al., 1992]. Polyhedrons in a 3D Voronoi diagram are guaranteed to be convex. This means a straight line in CIE LUV will never pass through a patch's Voronoi region more than once. Since the Voronoi regions of connected Munsell patches are themselves connected in CIE LUV, the names for colours along a line will change in a smooth, stable manner.

In order to assess the accuracy of our technique, we named known colour regions from the RGB, HSV, and CIE LUV colour models. We began with the RGB colour cube. Monitor RGB values from our Silicon Graphics workstations were converted to CIE XYZ, and then into CIE LUV to be named. The results are shown in Figure 3b. Colour names change smoothly along the cube's edges from corner to corner. The names provided appear to be correct, given that the monitor's RGB colour gamut covers only a subset of all visible colours (Figure 3a).

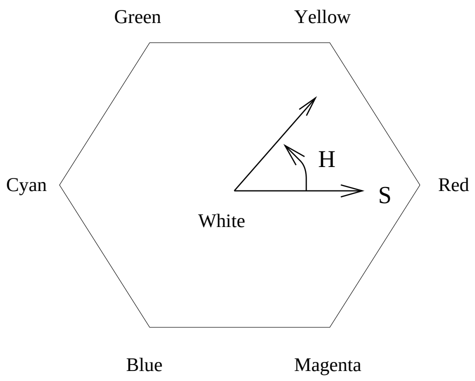
One apparent problem involves the RGB cube's white corner, which is labelled purple-blue rather than white. The Munsell patch matched to the white corner was 5PB 9/4, which is very close to white (Munsell patch 5PB 9/4 is two chroma "steps" from 5PB 9/0, since Munsell values from Wyszecki and Stiles are specified at even chroma values). A similar problem occurs during the naming of the cube's black corner (matched to Munsell patch 5GY 1/2). In the Munsell colour model, "white", "black", and "grey" are special cases, and are not used as hue names. We could solve this problem in certain situations by adding the colour names white, black, and grey. These correspond to Munsell colours with chroma zero and value ten (white), chroma zero and value zero (black), and chroma zero and value between one and nine (grey). Table 1 shows the Munsell patches and colour names we obtain for greys along the RGB cube's black-white diagonal.

The RGB cube's white corner is still named PB (although the black corner is now named "black"). This is because

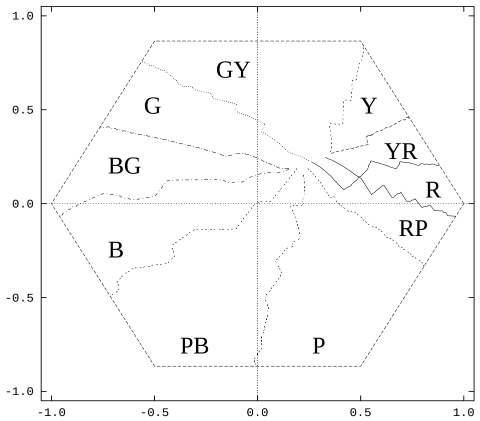
Monitor RGB	*Name*	*Munsell Patch*
(0.0, 0.0, 0.0)	Black	Black 0/0
(0.1, 0.1, 0.1)	B	5B 4/2
(0.2, 0.2, 0.2)	PB	5PB 5/2
(0.3, 0.3, 0.3)	B	5B 6/2
(0.4, 0.4, 0.4)	B	5B 7/2
(0.5, 0.5, 0.5)	B	5B 7/2
(0.6, 0.6, 0.6)	B	5B 8/2
(0.7, 0.7, 0.7)	PB	5PB 8/4
(0.8, 0.8, 0.8)	B	5B 9/2
(0.9, 0.9, 0.9)	PB	5PB 9/4
(1.0, 1.0, 1.0)	PB	5PB 9/4

Table 1: Names attached by our algorithm to colours along the black-white line in monitor RGB colour cube; colour constancy ensure that white, black, and grey values appear correct in the context of other monitor colours

the monitor's white corner is not in the same location as the white point in the Munsell colour model. Colour constancy makes the white value appear correct in the context of the other monitor colours. This last point also explains why corners of the colour cube are not named exactly as expected (*e.g.* both the yellow and the green corners are named GY). Our naming technique is in fact showing where the monitor's gamut falls within the region of visible colours.



(a)



(b)

Figure 4: Example of segmenting an HS-slice through the HSV hexcone at $V = 1$; (a) an HS-slice through the HSV hexcone, showing the primary colours at the corners of the hexcone, and the hue and saturation dimensions; (b) a segmented HS-slice at $V = 1$, lines represent the boundaries between the ten named categories

We continued investigating our naming technique by trying to categorize colours from a 2D slice through the HSV hexcone. The HSV colour model identifies colours using the three "dimensions" hue, saturation, and value. The HSV hexcone is a projection of the RGB colour cube along the white-black diagonal. This means the colour gamut of a monitor's HSV hexcone is the same as the gamut of the monitor's RGB cube.

The $V = 1$ plane of the hexcone contains a monitor's fully saturated red, green, blue, yellow, cyan, and magenta (Figure 4a). Our naming algorithm was used to partition this plane into the ten named categories shown in Figure 4b. These regions correspond closely to the names which are normally attached to the hexcone's corners. Each category boundary is well defined, and none of the boundaries intersect with one another. As with the RGB colour cube, the white point at the center of the hexcone falls within the purple-blue category. Our algorithm identifies what it would call white in the upper-right quadrant (somewhere in the region where the colour boundary lines converge). Again, this is due to the location of the monitor's gamut within the visible colour region.

We concluded our tests by trying to divide a 2D slice through the CIE LUV colour space. Unlike RGB and HSV, CIE LUV is not constrained by a set of monitor triad chromaticities. The CIE LUV colour model is capable of representing any colour from the visible frequency domain. We expected to see a much closer match between the white point in CIE LUV and the region our algorithm identifies as containing white, since the CIE explicitly calibrated CIE LUV to correspond closely to Munsell.

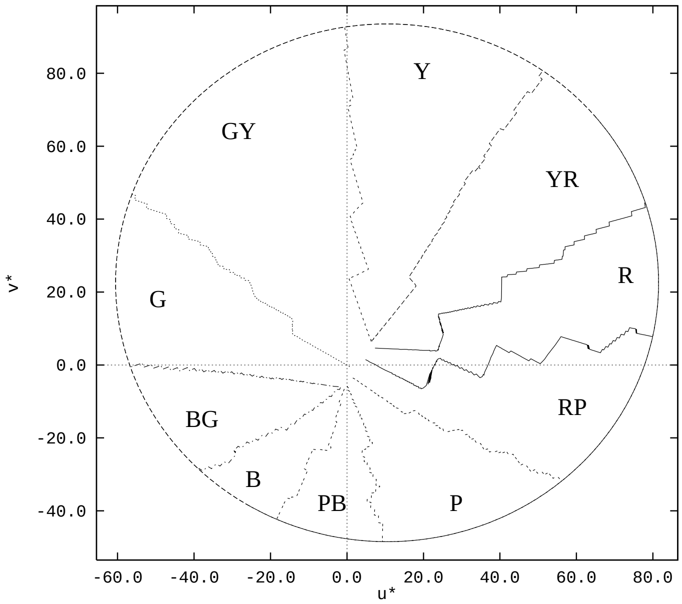


Figure 5: Example of segmenting a u^*v^*-slice through CIE LUV at $L^* = 71.6$; lines represent the boundaries between the ten named categories

Figure 5 shows a circular u^*v^*-slice through the CIE LUV model at a fixed luminance of $L^* = 71.6$. The circle was centered at $(u^*, v^*) = (10.49, 22.54)$ and had a radius of 71.0. The white point in CIE LUV is located at $(u^*, v^*) = (0, 0)$. As with the HSV example, each category boundary is clearly defined. Boundaries do not cross one another. Moreover, the colour boundary lines terminate as expected at the region which contains the CIE LUV white point.

4. EVALUATING COLOUR CATEGORIES

Results from our automatic segmentation algorithm suggest it is accurate and stable for a wide range of colour models and visible colours. Observers agreed that the names attached to individual colours were appropriate given the set of names available through the Munsell hue dimension. This does not mean that observers would have chosen exactly the same names used by the algorithm to describe each colour, however. For example, most people do not use the name "purple-blue" to describe a colour. They are more likely to use names like dark blue, grey blue, or even indigo.

The way in which users choose to name colours is important in situations where we want to control perceived colour difference. Many visualization techniques display different data values by varying colour [Ware and Beatty, 1988; Ware, 1988; Levkowitz and Herman, 1992; Bergman et al., 1995]. User-interface design is also concerned with the choice of colours for information representation [Robertson, 1988; Rheingans and Tebbs, 1990]. In order to perform these tasks effectively, it is important to understand how a user perceives colour difference. Using a perceptually balanced colour model can dramatically increase control over colour difference. However, research has shown that other factors like linear separation [D'Zmura, 1991; Bauer et al., 1996] or changes in colour category [Kawai et al., 1995] can also influence perceived colour difference.

In order to study these properties, we need a method for segmenting a colour gamut into user-named categories.

We used our algorithm to automatically divide the colour gamut into an initial starting state. From there, we ran a limited set of colour naming tests to try to determine the perceptual overlap, strength, and user-chosen name of each category.

4.1 Method

We decided to try to assess the categories along the circumference of the u^*v^*-circle shown in Figure 5. Our segmentation algorithm was used to divide the circle's boundary into ten initial categories. We needed to pick a representative colour from each category to use during our naming experiment. The colour at the center of the category was chosen for this purpose (Figure 6).

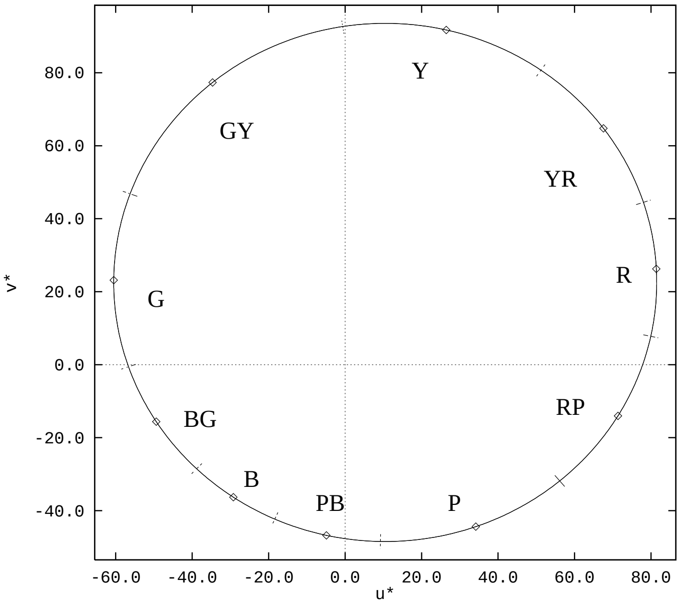


Figure 6: A u^*v^*-slice through CIE LUV at $L^* = 71.6$; ticks along the the circle mark the boundaries between the ten named categories, points are the representative colours for each category

Thirty-eight observers with normal or corrected acuity volunteered to participate in our experiment. Each observer was tested to ensure they were not colour blind. They were then asked to name each of the ten representative colours. Observers were told to give a simple name which accurately described each colour. No restrictions were placed on the names they were allowed to use.

Our experiment was conducted on a Macintosh computer with a sixteen inch colour monitor and video hardware which used an eight bit colour lookup table. The software used was designed and written specifically to run vision experiments [Enns and Rensink, 1991]. The chromaticities of the monitor's triads and maximum-intensity red, green, and blue were measured to build an accurate CIE LUV to monitor RGB conversion matrix.

During the experiment the representative colours were shown to each observer in a random order. Each colour was displayed as a rectangular patch which filled approximately 90% of the screen. The border around the outside of the patch was coloured light grey, and had a luminance $L^* = 71.6$ equal to the luminance of the colour patches. Observers were allowed to view the patch for as long as they wanted before providing a name. Observers were told that they would not be shown the same colour more than once. The names provided by each observer were recorded for later analysis.

4.2 Results

Observers provided us with a wide range of different colour names. We compressed these results by combining names which described the same colour or combination of colours. For example, the names "purple", "violet", and "mauve" from the P category were combined under the name "purple". The names "aqua", "turquoise", "blue green", "green blue", and "sea green" from the BG category were combined under the name "aqua". Figure 7 shows the names chosen for each of the ten representative colours. The frequency of each name is also provided. A name's frequency is the percentage of observers who chose that name to describe the given colour patch.

	purple	*magenta*	*pink*	*red*	*orange*	*brown*	*yellow*	*green*	*aqua*	*blue*	*other*
P	86.9%	2.6%	5.2%								5.2%
RP	15.7%	28.9%	55.3%								
R			26.3%	71.0%							2.6%
YR				5.3%	86.8%	7.9%					
Y					2.6%	44.7%	47.4%				5.2%
GY								97.3%			2.6%
G								100.0%			
BG								26.3%	57.8%	15.8%	
B									7.9%	89.4%	2.6%
PB	5.2%									92.1%	2.6%

Figure 7: Frequency of user-chosen names for each of the ten named category, represented by percentage of observers who chose each name to describe the given colour

Even a brief review of the results in Figure 7 leads to a number of interesting conclusions. For example, both the G and GY categories were almost exclusively named "green" (with frequencies of 100% and 97.4%, respectively). This suggests that, in terms of user-chosen names, G and GY should be collapsed into a single "green" category. The B and PB categories (both named "blue" with frequencies 89.4% and 92.1%, respectively) might also be collapsed into a single "blue" category. The R category appears to lack saturation; combining the frequencies for the names "red" and "pink" (unsaturated red) results in near-total coverage of 97.3%. Similarly, the Y category lacks luminance; a combination of the names "yellow" and "brown" (dull yellow) results in a total frequency of 92.2%.

Other categories appear to cover a range of user-chosen names. The RP category is split between "pink" (or unsaturated red, 55.3%), "purple" (15.7%), and "magenta" (or red-purple, 28.9%). Observers named this category using one or both of its primary components (either red or purple). The BG category was named either "green" (26.3%), "aqua" (or blue-green, 57.8%), or "blue" (15.8%). Observers were told after the experiment that the Munsell name for this colour was "blue-green". Most observers agreed that "blue-green" was a descriptive and accurate name (in spite of the fact that almost no one chose it for themselves). This suggests that Munsell names are not necessarily poor, rather they may be uncommon in most people's vocabulary.

4.3 Perceptual Overlap

Our results show clearly that certain neighbouring categories overlap in the names chosen to represent the category. Kawai suggest that a user's ability to detect a colour "target" will decrease dramatically when background elements use colours from the target's category. We could logically extend this to imply that a similar decrease in performance will occur if background colours are chosen from neighbouring but overlapping categories. In order to predict (and avoid) this type of visual interference, we need a method to measure perceived category overlap. This type of overlap depends on:

- the range of user-chosen names assigned to a given category
- the frequency of a user-chosen name
- how the ranges of neighbouring categories overlap

As an example, consider the R and YR categories, which have user-chosen name ranges:

	pink	*red*	*orange*	*brown*
R	26.3%	71.0%		
YR		5.3%	86.8%	7.9%

Colours from R and YR overlap only at the "red" name; their overlap is not that strong, since the frequency of "red" for the YR category is low. We defined the amount of overlap by multiplying the frequencies for the common name. This gives an R-YR overlap of $71.0\% * 5.3\% = 0.0376$. A closer correspondence of user-chosen names for a pair of categories results in a much larger overlap. For example, given the user-chosen name ranges for GY and G:

	green	*other*
GY	97.4%	2.6%
G	100.0%	

the GY-G overlap is $97.3\% * 100.0\% = 0.973$. Colours which overlap over multiple names are combined using addition, for example, BG and B have ranges:

	green	*aqua*	*blue*	*other*
BG	26.3%	57.8%	15.8%	
B		7.9%	89.4%	2.6%

for a BG-B overlap of $(57.8\% * 7.9\%) + (15.8\% * 89.4\%) = 0.187$. When we use this algorithm to compute the overlap between each of the ten named categories, we obtain the overlap table shown in Figure 2. Values from this table can be used to predict when certain colours might be difficult to distinguish from one another.

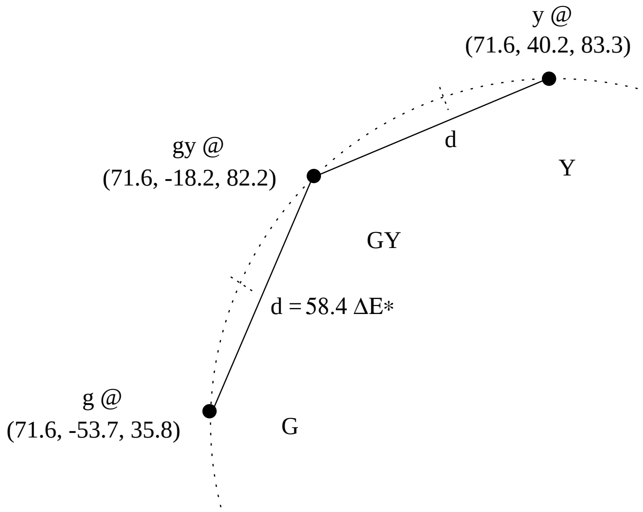
	R	YR	Y	GY	G	BG	B	PB	P	RP
R	—	.038	0	0	0	0	0	0	.014	.145
YR	.038	—	.058	0	0	0	0	0	0	0
Y	0	.058	—	0	0	0	0	0	0	0
GY	0	0	0	—	.973	.256	0	0	0	0
G	0	0	0	.973	—	.263	0	0	0	0
BG	0	0	0	.256	.263	—	.187	.146	0	0
B	0	0	0	0	0	.187	—	.823	0	0
PB	0	0	0	0	0	.146	.823	—	.045	.008
P	.014	0	0	0	0	0	0	.045	—	.173
RP	.145	0	0	0	0	0	0	.008	.173	—

Table 2: Table showing the degree of user-name overlap between representative colours from each of the ten named categories

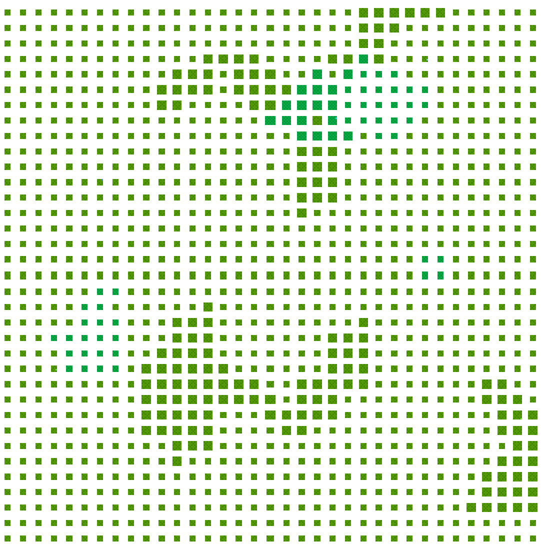
5. DATA VISUALIZATION USING COLOUR

To show how our results might be used during visualization design, assume we wanted to encode a binary (*i.e.* two-valued) data attribute using hue. One possible choice of colours is (g, gy) where $g \in$ G category, $gy \in$ GY category, and the distance (or perceived difference) between g and gy in CIE LUV is d. A second choice is (y, gy), that is, we use $y \in$ Y instead of $g \in$ G. As in the first case, distance in CIE LUV between gy and y is fixed to be d (Figure 8). We expect the first case to give poor colour differentiation relative to the second case (in spite of a constant colour distance d between the colours pairs in CIE LUV). This is because users perceive both g and gy as part of the same "green" colour category.

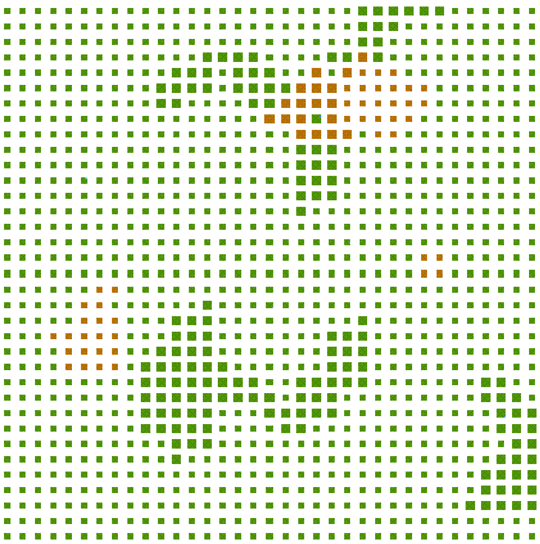
Initial experiments now being run in our laboratory have confirmed our expectations. Colours chosen from categories with a large perceptual overlap (like G and GY) are difficult to differentiate from one another. Colours from categories with a low perceptual overlap (like GY and Y) give better perceptual contrast and correspondingly better



(a)



(b)



(c)

Figure 8: (a) Two possible colour pairs (g, gy) and (y, gy) chosen from the G, GY, and Y colour categories; distance between each pair of colours in CIE LUV is $58.4\,\Delta E^*$; in spite of the perceptual balance provided by CIE LUV, results show that (g, gy) are harder to distinguish from one another compared to (y, gy); (b) the number (three) and boundaries of the g regions are difficult to determine relative to; (c) the same regions of interest encoded with y rather than g

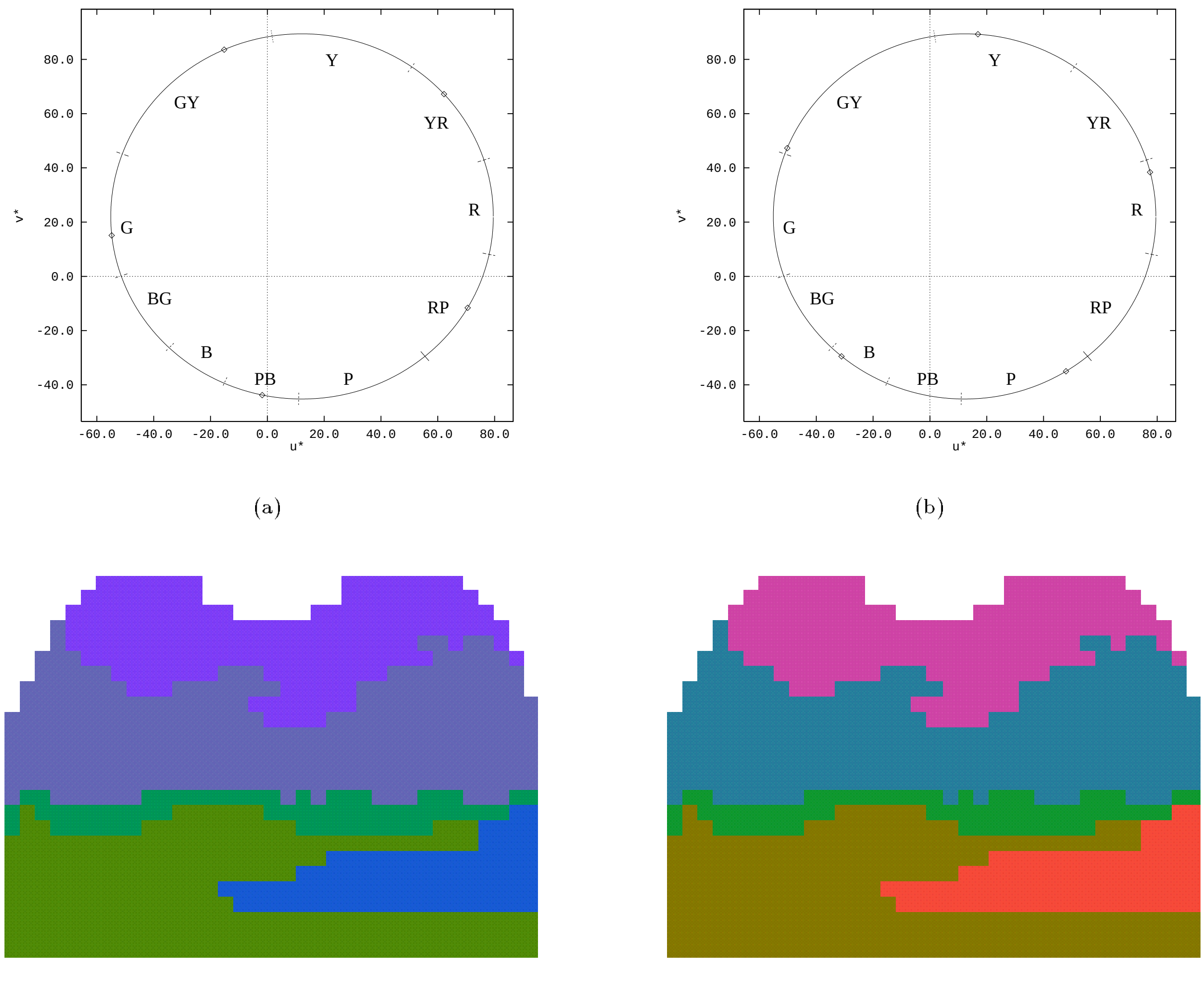


Figure 9: Two possible five-colour sets used to display five temperature regions on a regular grid, distance between any two pairs of colours is fixed to $d = 79.1\,\Delta E^*$, all colours are isoluminant with $L^* = 71.6$: (a) five colours from the YR, GY, G, PB, and RP categories represented as dots; (b) five colours from the R, Y, GY, B, and P categories represented as dots; (c) first five-colour set, users have difficulty determining the location and boundaries of the thin temperature region (running horizontally through the middle of the grid) coloured using $g \in$ G, since it borders on a $gy \in$ GY region; (d) second five-colour set, users can more easily detect the thin temperature region in particular, and the boundaries of all regions in general since none of the colours' categories overlap

performance during exploratory visualization tasks. This it true even when colour distance between the colours is fixed to a constant value in a perceptually uniform colour model.

Obviously if we were encoding a binary data attribute with colour, we would pick two colours which were clearly distinguishable from one another (*e.g.* $g \in$ G and $r \in$ R). This becomes much more difficult to do when the attribute being encoded is multi-valued. For example, given an attribute with five possible values, we need to pick five colours which are all equally distinguishable from one another. In this case, it may no longer possible to locate all the colours a large distance away from one another. Additional colour selection effects may also restrict the region of available colours. For example, colours might need to be isoluminant with one another, since research by Callaghan suggests that random variations in intensity interfere with visual tasks performed using colour [Callaghan, 1984]. Understanding how neighbouring categories overlap is much more important, since some (or all) of the colours will almost certainly be chosen from adjacent categories (Figure 9). Values from our overlap table can be used to help ensure that colours are not chosen from categories with a high perceptual overlap.

A final characteristic which we measured was the strength of each colour category. We defined a category to be strong when:

- it has one high frequency user-chosen name
- it has relatively low perceptual overlap with neighbouring categories

Using this definition and results for categories along our CIE LUV colour wheel, we classified YR (with a total perceptual overlap of 0.096) and P (with an overlap of 0.232) as being strongly named "orange" and "purple". The G-GY categories might be collapsed into a single strong "green" category (overlapping somewhat with BG). Similarly, B-PB might be collapsed into a single strong "blue" category (again overlapping somewhat with BG). Although categories like R and YR have low perceptual overlap, there was no common agreement on a single name to describe them. Users were evenly split between "red" and "pink" for the R category, and "yellow" and "brown" for the Y category.

5. DISCUSSION

This paper presents a simple technique for automatically segmenting colour regions into categories closely associated with Munsell hue names. Our algorithm takes into account perceptual colour difference by performing its matching in CIE LUV. We use Euclidean distance to find the nearest Munsell patch; this guarantees that boundaries between colour names are stable and well defined. Our technique accurately divided known colour regions from RGB, HSV, and CIE LUV.

One potential concern is the set of colour names provided by the Munsell colour model. Perceived colour difference and colour identification require user agreement for the names attached to individual colours. We provided a simple psychological method to assess the perceptual overlap and strength of each named category. During the experiment we picked representative colours from ten named categories along a circle in CIE LUV. Observers were asked to name each colour. Results allowed us to compute a measure of perceptual overlap between pairs of categories. These values can be used during the design of colour visualization techniques. Both past research and results from our own laboratory show that colours from categories with a high perceptual overlap are often difficult to distinguish from one another (relative to colours the same distance apart in a perceptual colour model, but chosen from non-overlapping categories). Finally, we define strong categories to be those with low perceptual overlap and a single high frequency user-chosen name.

During our experiments we found certain categories appeared to lack saturation or luminance (*e.g.* RP and Y, respectively). The Munsell model has two dimensions chroma and value which correspond directly to saturation and luminance. Our automatic segmentation technique could be modified to take these values into account. This would provide a finer division of colour regions into a larger number of colour categories. Additional design and experimentation is needed to decide where to place saturation and luminance category boundaries within the Munsell model.

Another extension to our algorithm which might help address the above problem is the use of CNS or CNM names. In particular, CNM "renames" Munsell into a much larger set of representative colour names. Using these names to divide colour gamuts might results in fewer weak categories. One potential drawback is the number of named categories for a given colour gamut. This would increase proportional to the increase in colour names. The CNM accuracy levels might be used to control the tradeoff between the strength of a named category and the total number of categories for a particular colour gamut.

REFERENCES

Bauer, B., Jolicoeur, P., and Cowan, W. B. 1996. Visual search for colour targets that are or are not linearly-separable from distractors. *Vision Research* (in press).

Bergman, L. D., Rogowitz, B. E., and Treinish, L. A. 1995. A rule-based tool for assisting colormap selection. In *Proceedings Visualization '95*, 118–125, Atlanta, Georgia.

Berk, T., Brownston, L., and Kaufman, A. 1982. A new colour-naming system for graphics languages. *IEEE Computer Graphics & Applications 2*, 3, 37–44.

Birren, F. 1969. *Munsell: A Grammar of Color*. Van Nostrand Reinhold Company, New York, New York.

Callaghan, T. C. 1984. Dimensional interaction of hue and brightness in preattentive field segregation. *Perception & Psychophysics 36*, 1, 25–34.

CIE 1976. *CIE Publication No. 15, Supplement Number 2 (E-1.3.1): Official Recommendations on Uniform Color Spaces, Color-Difference Equations, and Metric Color Terms*. Commission Internationale de L'Èclairge.

D'Zmura, M. 1991. Color in visual search. *Vision Research 31*, 6, 951–966.

Enns, J. T. and Rensink, R. A. 1991. VSearch Colour: Full-colour visual search experiments on the Macintosh II. *Behavior Research Methods, Instruments, & Computers 23*, 2, 265–272.

Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. 1990. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Kaufman, A. 1986. Computer artist's colour naming system. *The Visual Computer 2*, 4, 255–260.

Kawai, M., Uchikawa, K., and Ujike, H. 1995. Influence of color category on visual search. In *Annual Meeting of the Association for Research in Vision and Ophthalmology*, paper #2991, Fort Lauderdale, Florida.

Levkowitz, H. and Herman, G. T. 1992. Color scales for image data. *IEEE Computer Graphics & Applications 12*, 1, 72–80.

Miyahara, M. and Yoshida, Y. 1988. Mathematical transform of (R,G,B) colour data to munsell (H,V,C) colour data. In *Visual Communications and Image Processing '88*, 650–657. SPIE.

NBS 1976. *Color: Universal Language and Dictionary of Names*. National Bureau of Standards, special publication 440 edition.

Okabe, A., Boots, B., and Sugihara, K. 1992. *Spatial Tesselalations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Inc., New York, New York.

Rheingans, P. and Tebbs, B. 1990. A tool for dynamic explorations of color mappings. *Computer Graphics 24*, 2, 145–146.

Robertson, P. K. 1988. Visualizing color gamuts: A user interface for the effective use of perceptual color spaces in data displays. *IEEE Computer Graphics & Applications 8*, 5, 50–64.

Tominaga, S. 1985. A colour-naming method for computer color vision. In *Proceedings of the 1985 IEEE International Conference on Cybernetics and Society*, 573–577, Tucson, Arizona.

Ware, C. 1988. Color sequences for univariate maps: Theory, experiments, and principles. *IEEE Computer Graphics & Applications 8*, 5, 41–49.

Ware, C. and Beatty, J. C. 1988. Using colour dimensions to display data dimensions. *Human Factors 30*, 2, 127–142.

Wyszecki, G. and Stiles, W. S. 1982. *Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd Edition.* John Wiley & Sons, Inc., New York, New York.