

# Generating Color Palettes using Intuitive Parameters

Martijn Wijffelaars<sup>1,2</sup>, Roel Vliegen<sup>1</sup>, Jarke J. van Wijk<sup>2</sup> and Erik-Jan van der Linden<sup>2</sup>

<sup>1</sup>MagnaView BV, Eindhoven, the Netherlands  
<sup>2</sup>Eindhoven University of Technology, the Netherlands

---

## Abstract

*Color is widely used in data visualization to show data values. The proper selection of colors is critical to convey information correctly. In this paper, we present a technique for generating univariate lightness ordered palettes. These are specified via intuitive input parameters that are used to define the appearance of the palette: number of colors, hue, lightness, saturation, contrast and hue range. The settings of the parameters are used to generate curves through CIELUV color space. This color space is used in order to correctly translate the requirements in terms of perceptual properties to a set of colors. The presented palette generation method enables users to specify palettes that have these perceptual properties, such as perceived order, equal perceived distance and equal importance. The technique has been integrated in MagnaView, a system for multivariate data visualization.*

Categories and Subject Descriptors (according to ACM CCS): <http://www.acm.org/class/1998/> I.3.8 [Computer Graphics]: Applications

---

## 1. Introduction

Color is used in all areas of visualization, including cartography, information visualization and scientific visualization. The prime application is to encode data, but it is also used to highlight certain aspects or for aesthetical purposes. In this paper, we focus on the use of color for data encoding, and especially for ordered univariate data. Here, a scalar value is mapped to a color, either by using a continuous function or via a discrete mapping. The resulting ordered set of colors is called a *palette*. Selecting the right combination of colors in a palette is critical for conveying information correctly.

This task is not trivial. A number of sometimes conflicting requirements have to be met, moreover, the functional characteristics of the human visual system have to be taken into account. Experts can achieve optimal results by carefully hand-picking colors; for non-experts this is often too difficult and time-consuming. A simple solution is to provide the user with a number of standard palettes. However, none of these may be deemed optimal for the application under consideration, and hence a custom palette has to be created.

In this paper, we propose a model for the automated synthesis of palettes. Users are provided with a limited set of intuitive parameters, which they can use to quickly gener-

ate a wide variety of proper palettes. We attempt to produce palettes with a similar quality as carefully manually designed palettes.

In section 2 we present an overview of manually designed palettes and methods for creating palettes. In section 3 we formulate our objectives. In section 4 we present our palette generation model. Colors follow from sampling a curve through the CIELUV color space. In section 5 we show the integration of the technique in the visualization application MagnaView and present the results of our user study. Finally, the results are discussed and possibilities for future work are presented in section 6.

## 2. Related work

There are three main requirements for proper univariate ordered palettes [Tru81, Lev96, Bre99]:

1. The colors display a perceived ordering;
2. The perceived color distances are representative of the intended distances;
3. The colors are perceived as equally important.

An example of a palette that fails in all these aspects is the widely used rainbow color map [BT07]. Besides these general requirements, other visualization-specific requirements

have to be satisfied. For instance, red roads on a red background are invisible, or, perceptual steps between consecutive colors need to be higher if the spatial frequency of the visualization is high.

### 2.1. Brewer palettes

The best results are produced by experts that carefully select the most appropriate colors. Prime examples are the palettes designed by Cynthia Brewer [Bre99, Bre05], which are often used and highly respected in the visualization community. We therefore consider them as reference prototypes for automatically generated palettes.

The Brewer set of palettes is large, but finite. For a number of hues no palettes are provided, and also, palettes with a large number of colors up to continuous palettes are not supported. Furthermore, deviations from these palettes might be needed such that the palette colors do not interfere with the other colors used in the visualization. To provide a larger set of palettes, automated generation is required.

### 2.2. Color systems

Many visualization toolkits have implemented some method for palette generation. These methods are mostly based on the RGB, HSV (also known as HSB) or HLS (also known as HSL and HSI) color systems. In these color systems paths are defined, mostly via lines or curves, along which the palette colors are sampled. One of the most sophisticated in this category is PRAVDAColor [BRT95].

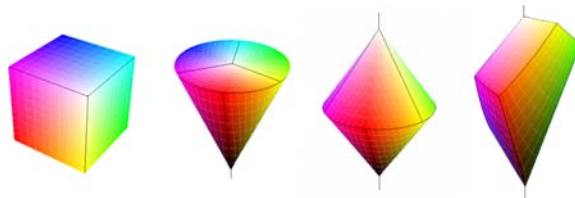
These color systems can be used to generate palettes that display a perceived ordering, but the perceived color distances are not representative for the intended distance. This requires a color system that is perceptually uniform, i.e., colors having equal values in certain dimensions should have equal perceptual properties in that dimension. As an example, two colors of lightness value 50 are perceived as equally light. Figure 1 shows examples of non perceptual uniformity of the HLS space.



**Figure 1:** Examples of non perceptual uniformity in HLS. Left: Three colors of equal lightness. Middle: Two pairs of colors with equal hue distance. Right: Two pairs of colors with equal lightness distance.

A solution is to use a color system based on human perception, such as CIELUV or CIELAB [cie86]. The CIELUV system is recommended in additive light source conditions, the CIELAB system for use under reflected light conditions

[RO86]. This, however, introduces a new problem. These color spaces use device independent colors, therefore care must be taken to use only colors that fall within the gamut of the used device, i.e., within displayable color space (Figure 2).



**Figure 2:** The RGB cube, the HSV cone, the HLS double-cone and displayable CIELUV space

### 2.3. Sampling color space

Others have used the CIELUV or CIELAB color space as the basis of a palette generation method [Lev96, RO86, ZH06, ZM06, Pha90]. Some use sampled straight lines to define their palettes. However, then the irregularly shaped color spaces can not be used to their full potential. For instance, the only way to achieve maximum lightness contrast with a straight line is to go from black to white, creating a gray scale. Some use line segments to solve this. This, however, introduces discontinuities in the first order derivative of the path. The sharper the bends in the path, the more attention is drawn towards these discontinuities, thereby violating the equal importance requirement.

All palette generation methods referred to above use color keys to specify their palettes. To generate a proper palette, i.e., one that fits the requirements stated in the beginning of section 2, the color keys have to be chosen carefully. Therefore, the user is required to have experience in the field of creating color palettes. This is reflected in a recent paper by Zhang and Montag [ZM06] which states:

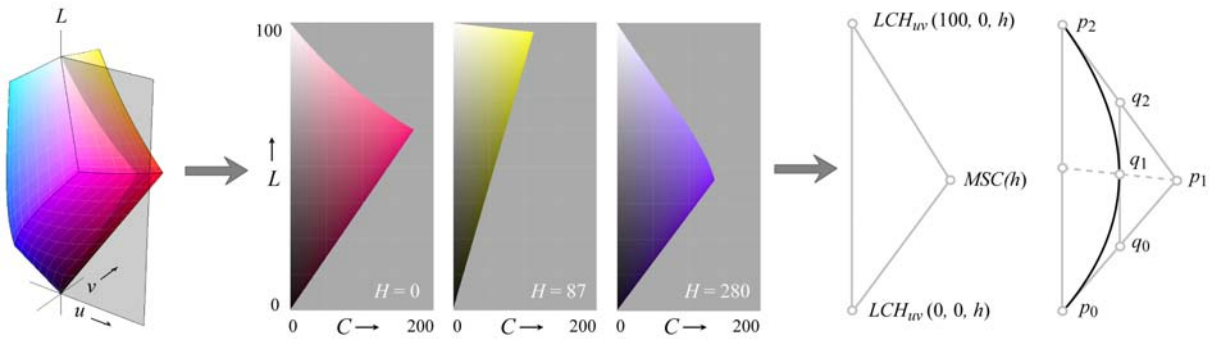
"The construction of color schemes is a subtle task and the design process is mostly ad hoc."

Furthermore, they give no solution on how to prevent the path from exiting displayable color space.

## 3. Overview

Our aim is to aid the user such that he can only generate palettes that satisfy the given requirements, while still being able to fine-tune a palette to a visualization and to his preferences.

Our solution is to use a curve instead of line segments. This way, it is possible to utilize the complete lightness range and have no abrupt bends in the path. Furthermore, we propose to help the user by specifying palettes using a limited set of intuitive parameters, rather than color keys. The first three are *hue*, *saturation* and *brightness*, in accordance with



**Figure 3:** The volume of displayable colors in CIELUV space, three hue slices of CIELUV space generated with PaletteView and the triangle that approximates the boundary of the displayable colors of a hue slice

the intuitive dimensions of color spaces. Further parameters are the *number of colors* of the palette, the *cold/warm* parameter that controls the perceived color temperature, and the *contrast* parameter that defines the overall lightness contrast of the palette. These parameters make it possible to fine-tune palettes to visualization-specific requirements; for instance, a palette in a certain lightness range ensures that white text is always clearly visible, or, using a palette with lowly saturated colors makes it possible to highlight other aspects of the visualization with saturated colors.

#### 4. Model

In this section, we give a short introduction to the CIELUV color space, then analyze Brewer palettes within that space. Based on this analysis, we propose a model for generation of single-hue palettes, which is extended to a multi-hue model.

##### 4.1. CIELUV

The CIEXYZ system forms the basis of CIELUV. The XYZ coordinates of a given color stimulus can be calculated by measuring its spectral power distribution with a spectrophotometer. Via multiplication with an RGB-matrix, the XYZ coordinates can be converted to RGB and back.

The XYZ space is not perceptually uniform. To make it more uniform, CIEXYZ is transformed to CIELUV. Via a further transformation, CIELUV can be converted to LCH<sub>uv</sub>, yielding the intuitive dimensions lightness, saturation and hue. LCH<sub>uv</sub> makes it possible to specify a color in both perceptual and meaningful terms. For more information on the conversion of colors between color systems we refer to [cie86, Lin].

During the development and testing of our palette generation technique CRT monitors were used, since they have superior color reproduction to LCD displays. However, the color palettes can be calibrated to other monitors by changing parameters in the color space conversion formulae, such as gamma, reference white, and the RGB-matrix.

It is possible to specify a color in LCH<sub>uv</sub> outside the displayable color space of a monitor, i.e., if its corresponding RGB value is not in the range of  $[0..1]^3$ . By clamping the RGB components back to that range, a displayable color is obtained. We aim to use LCH<sub>uv</sub> values that are within displayable color space, to avoid the irregularities that result from color clamping.

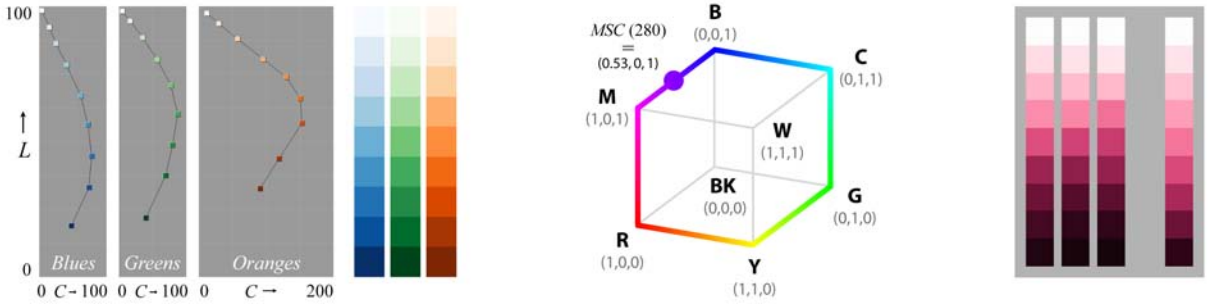
Figure 2 (right) shows the RGB-cube, representing the displayable colors, transformed to CIELUV space. To investigate color spaces, we developed an application called PaletteView. This application generates slices from color spaces shown in four separate viewports. Figure 3 contains three such slices, each showing the displayable colors of a single hue. Note that each hue slice has a different shape, in contrast with hue slices of HSV or HLS. The CIELUV slices better match our perception, because the most saturated colors of each hue reside at different lightness levels; yellow colors are saturated at high lightness values, blue colors at low lightness values. The example colors in Figure 1 that have equal lightness in HLS, have a lightness value of 60, 97 and 46 in CIELUV space.

We use the convention  $LCH_{uv}(60, 20, 270)$  to denote a color of a certain color space, in this case a LCH<sub>uv</sub> color with lightness 60, saturation 20 and hue 270. To convert a color  $c$  to a certain color space we use the notation  $RGB(c)$ , which gives the RGB coordinates of  $c$ . A color component is obtained by adding a suffix, e.g.,  $c.L$  yields the lightness of  $c$ .

##### 4.2. Brewer palettes in CIELUV

To analyze the Brewer palettes within the CIELUV color space, the RGB coordinates of their colors are converted to LCH<sub>uv</sub> space. Figure 4 (left) shows three single-hue palettes of Brewer, with lightness on the vertical axis and saturation on the horizontal axis. Based on the plots, a number of observations can be made:

- As the palettes increase in lightness, the saturation increases and then decreases. This indicates that for a clear



**Figure 4:** *Left:* Three Brewer palettes plotted in  $LCH_{uv}$  space with PaletteView. *Center:* The RGB cube with all MSCs. *Right:* Colors at equal distance according to  $\Delta E_{uv}^*$ , CIE94, CIEDE2000, and the L function.

order to be perceived a monotonic saturation increase or decrease is not necessary. We observed that as long as the saturation changes are not too abrupt, they do not violate the order or the equal importance requirements.

- The palettes do not increase linearly in CIELUV lightness; the lighter colors are closer to each other than the darker colors.
- The Brewer palette paths have saturation peaks at a different lightness levels. The shape of the path follows from the shapes of the hue slices in CIELUV space.

#### 4.3. Displayable colors of a hue

Our approach for palette generation is to define a continuous path through color space, sampled at uniform color distance intervals, to obtain a discrete palette with finite numbers of colors.

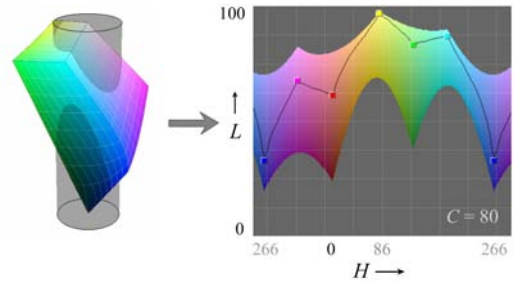
We first consider the space available for a path of a single hue. A hue slice of displayable color space in CIELUV can be characterized by three points: white, black and the  $MSC(h)$ , i.e., the *Most Saturated Color* of a hue. The  $MSC$  is not at equal lightness and saturation for each hue (Figure 3). The triangular surface spanned by black, white and the  $MSC$ , is an approximation of the displayable color space of a certain hue. Using this triangular surface, only a small portion of the displayable color space is not used, and little clamping is required to make all colors displayable.

All  $MSC$ s lie on edges of the RGB cube characterized by one RGB component at value 0, one at value 1, and one variable component (Figure 4 center). To calculate the  $MSC$  of a given hue  $h$ , first the correct edge is determined, next the intersection point between this edge and the CIELUV hue plane belonging to  $h$  is calculated. In Figure 4 (center), for  $MSC(280)$  holds that  $G = 0$ ,  $B = 1$ , next  $R$  can be calculated by

$$R = \left( -\frac{(\alpha u_n + \beta v_n)(m_{BX} + 15m_{BY} + 3m_{BZ}) - (4\alpha m_{BX} + 9\beta m_{BY})}{(\alpha u_n + \beta v_n)(m_{RX} + 15m_{RY} + 3m_{RZ}) - (4\alpha m_{RX} + 9\beta m_{RY})} \right)^{\frac{1}{\gamma}},$$

where  $\alpha = -\sin(h)$  and  $\beta = \cos(h)$  [Wij08]. Furthermore,

$\gamma$  stands for the gamma value,  $u_n$  and  $v_n$  denote the  $(u, v)$  coordinate of the reference white, and  $m_{ij}$  denotes a value of the RGB-matrix.



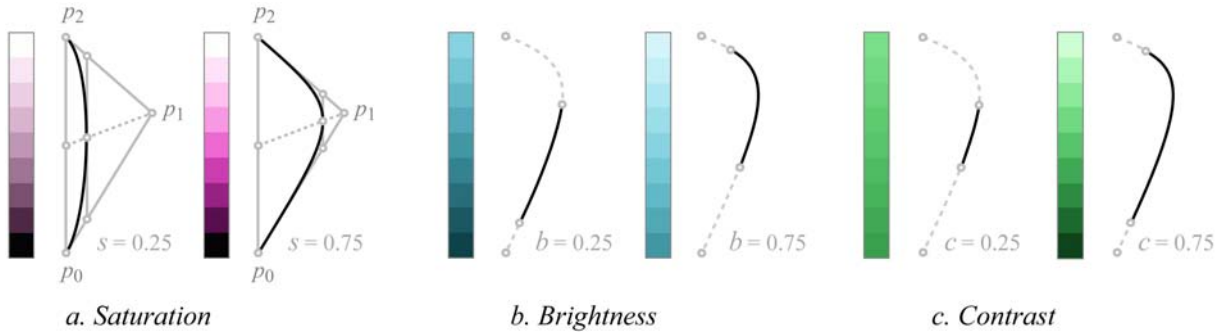
**Figure 5:** *Left:* the CIELUV displayable color volume with a cylinder that defines all colors of similar saturation. *Right:* the corresponding saturation slice generated with PaletteView. The path of the  $MSC$ s of all hues is shown, indicating the lightness of each  $MSC$ .

#### 4.4. Single-hue sequential palette model

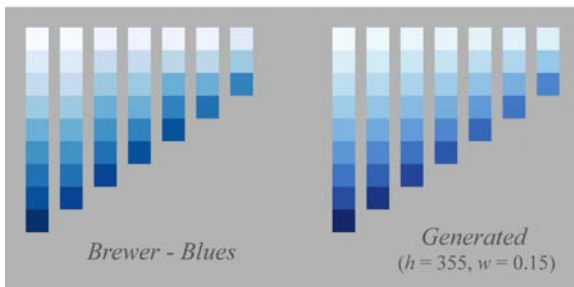
The triangular surface that approximates the displayable colors of a given hue, is used as the basis for the path on which all colors of a palette are sampled. The path can be specified using the following intuitive control parameters:

- $N$ : number of colors, ( $N = \infty$  for continuous palettes);
- $h$ : the main hue;
- $c$ : the overall lightness contrast;
- $s$ : saturation;
- $b$ : brightness.

For reference the complete single-hue sequential palette model is given in table 1. The points  $p_0$ ,  $p_1$  and  $p_2$  define the triangular surface, defined by  $LCH_{uv}(0, 0, h)$ ,  $MSC(h)$  and  $LCH_{uv}(100, 0, h)$ . We define the continuous curve  $C_{seq}$  as two quadratic Bézier curves using  $(p_0, q_0, q_1)$  and  $(q_1, q_2, p_2)$  as control points (Figure 3). This yields a path from black, through a range of colors of equal hue, to white.



**Figure 6:** Changing the curve with saturation, brightness and contrast parameters



**Figure 7:** Left: seven palettes made by Brewer. Right: seven generated palettes with varying contrast

The saturation parameter  $s$  controls the tension of curve  $C_{seq}$ ; if  $s = 0$  then  $C_{seq}$  is a straight line from  $p_0$  to  $p_2$ , if  $s = 1$ ,  $C_{seq}$  is a path from  $p_0$  to  $p_1$  to  $p_2$  (Figure 6). The brightness parameter  $b$  determines the starting position of the curve. The lightness contrast parameter  $c$  determines the length of the curve (Figure 6). In the Brewer palettes the lightness contrast is related to the number of colors in the palette. The default value for  $c$ ,  $c = \min(0.88, 0.34 + 0.06N)$ , approximates this relation (Figure 7).

#### 4.5. Distance function

In an ideal perceptually uniform model, colors at equal parameter distance have an equal perceived distance. Although CIELUV and CIELAB are better than the simpler models, still distance functions are used to calculate the perceived distance between two colors. The CIE has defined several distance functions:  $\Delta E_{uv}^*$ , CIE94 and CIEDE2000 [cie86, Lin]. However, neither of these yielded satisfactory results, which is shown in Figure 4 (right). Therefore, it is necessary to introduce a new distance function  $L$ , for use with our palette definition. Based on the colors of the Brewer palettes in Figure 4 (left),  $L$  transforms the color space in such a way that palette colors are closer to each other in CIELUV when their lightness is high, and further apart when their lightness is low. The  $L$  function considers only the L component of the colors, since lightness is the dominant perceptual compo-

**Table 1:** Single-hue sequential palette model.

$P_{seq}[i] = P_{seq}(\frac{i}{N-1}),$	$i \in [0..N-1]$
$P_{seq}(t) = C_{seq}(T(L(t))),$	$t \in [0..1]$
$L(t) = 125 - 125 \cdot 0.2^{(1-c)b+tc},$	$t \in [0..1]$
$C_{seq}(t) = \begin{cases} B(p_0, q_0, q_1, 2t), & t \leq 0.5 \\ B(q_1, q_2, p_2, 2(t-0.5)), & t > 0.5 \end{cases}$	
$T(\ell) = \begin{cases} 0.5B^{-1}(p_0.L, q_0.L, q_1.L, \ell), & \ell \leq q_1.L \\ 0.5B^{-1}(q_1.L, q_2.L, p_2.L, \ell) + 0.5, & \ell > q_1.L \end{cases}$	
$B(b_0, b_1, b_2, t) = (1-t)^2 b_0 + 2(1-t)t b_1 + t^2 b_2$	
$B^{-1}(b_0, b_1, b_2, v) = \frac{b_0 - b_1 + \sqrt{b_1^2 - b_0 b_2 + (b_0 - 2b_1 + b_2)v}}{b_0 - 2b_1 + b_2}$	
$p_0 = LCH_{uv}(0, 0, h)$	$q_0 = (1-s)p_0 + sp_1$
$p_1 = MSC(h)$	$q_1 = \frac{1}{2}(q_0 + q_2)$
$p_2 = LCH_{uv}(100, 0, h)$	$q_2 = (1-s)p_2 + sp_1$
Default values:	
$s = 0.6, b = 0.75, c = \min(0.88, 0.34 + 0.06N)$	

nent that produces an ordering in a sequential palette. Points on curve  $C_{seq}$ , which is comprised of two quadratic Bézier curves, can now be found analytically. First the lightness of a color is calculated with  $L(t)$ , since  $C_{seq} \cdot L$  now is equal for all parameter settings. Then,  $T$  is used to calculate the curve parameter value that belongs to this lightness. Using this value the corresponding saturation and hue are calculated with  $C_{seq}$ .

#### 4.6. Multi-hue and diverging palettes

Multi-hue palettes make it easier to detect to which cluster a certain color belongs, because of the gradual change in hue [SKR99, Lev96, BT07]. By analyzing the Brewer palettes with PaletteView, we found out that many palettes



move towards yellow as they get lighter, even most single-hue palettes. A reason for this is that yellow colors can be saturated at a high lightness (Figure 5). We make use of this, when generating multi-hue palettes, by shifting the top point  $p_2$  towards a bright point  $p_b$ , defined as yellow. Parameter  $w$  controls how close the top point  $p_2$  is to  $p_b$ . However, the resulting point could lie outside the triangular surface that approximates displayable colors. Therefore,  $S_{max}$  calculates the maximum saturation for a color of a certain lightness and hue; it projects the color back to the triangular surface, and preserves its lightness and hue value. The hues are interpolated using the shortest path on the color circle, for instance, mixing hues 60 and 340 yields 20. Since moving colors towards yellow makes their appearance warmer and moving them away colder, we call the  $w$  parameter *cold/warm*. Table 2 shows the multi-hue model as an extension to the single-hue model of Table 1. With this model, a wide range of multi-hue palettes can be generated.

Brewer also specifies diverging palettes. A diverging palette is created by concatenating two sequential palettes, with a combined neutral point. The neutral point is white or gray in case of single-hue palettes and moves towards the bright point  $p_b$  as  $w$  increases for multi-hue palettes. Figures 9 and 10 show examples of generated diverging palettes.

**Table 2:** Multi-hue sequential palette model.

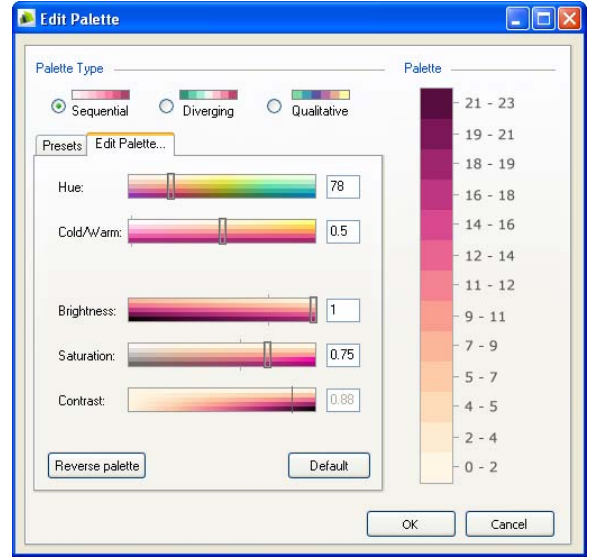
$$\begin{aligned}
 p_2.L &= 100(1 - w) + w \cdot p_b.L \\
 p_2.S &= \min(S_{max}(p.L, p.H), w \cdot s \cdot p_b.S) \\
 p_2.H &= (h + \alpha M) \bmod 360 \\
 M &= (180 + p_b.H - h) \bmod 360 - 180 \\
 S_{max}(l, h') &= \alpha(p_{mid}.S - p_{end}.S) + p_{end}.S \\
 \text{where} \\
 \alpha &= (p_{end}.L - l) / (p_{end}.L - p_{mid}.L) \\
 p_{mid} &= MSC(h') \\
 p_{end} &= \begin{cases} LCH_{uv}(0, 0, h'), & l \leq p_{mid}.L \\ LCH_{uv}(100, 0, h'), & l \geq p_{mid}.L \end{cases} \\
 \text{Default values:} \\
 p_b &= LCH(1, 1, 0), \quad w = 0
 \end{aligned}$$

## 5. Results

We have implemented our palette generation technique in the visualization application MagnaView [Mag], that visualizes multivariate data using generalized treemaps [VvWvdL06]. Each tile in the treemap is assigned a color based on a mapping between a data attribute and a color palette.

*Novel users* can select preset palettes that we defined using the parameters hue, lightness, saturation, contrast and

hue range (Figure 10). The number of colors is derived from the data. The type of data is used for selecting between sequential and diverging palettes; for instance, unsigned numerical data is assigned a sequential palette and signed numerical data a diverging palette.

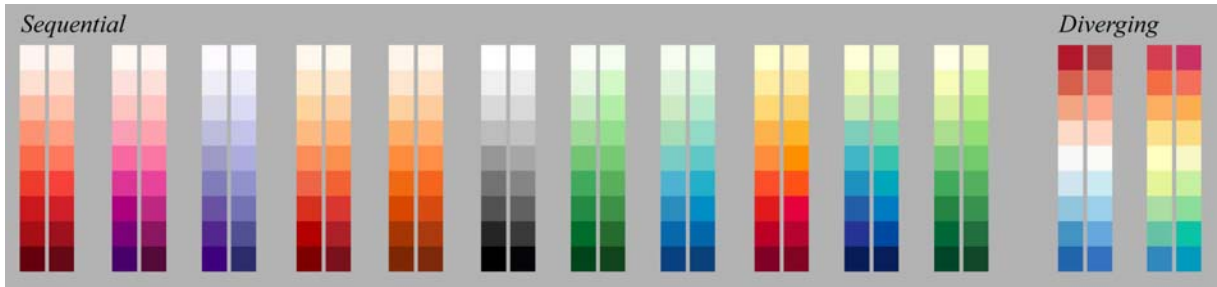


**Figure 8:** User interface for editing palettes in MagnaView.

*Experienced users* can alter all parameters of the (preset) palettes using sliders that show the possible palette appearances for all values of a dimension (Figure 8). This way, the user can see beforehand how changing a certain parameter will affect the palette. Because it is important to see a set of colors in context [MSK04], our aim is that the user can see the colors in the visualization change when the palette is changed; however, we have not implemented this yet. Our palette generation technique can only generate palettes that satisfy the three palette requirements, therefore users can only create proper palettes, while still having a wide range of possibilities. If needed, *expert users* can still set all colors individually.

### 5.1. User Study

We performed an informal user study that consisted of two parts. The first part was intended to evaluate the slider interface described above and the second part to test the aesthetic qualities of the generated palettes. The latter was included in order to find the most appealing default palettes for MagnaView. Fifteen test subjects with normal color vision participated in the user study, of whom seven were MagnaView employees and eight were individuals who had never used MagnaView before. In the first part of the user study, the test subjects were given a short introduction to the MagnaView application. Next, they were given five tasks that each corresponded to a different visualization. The tasks



**Figure 9:** Pairs of palettes of which the left palettes are defined by Brewer, and the right palettes are generated with  $P_{seq}$ .

ranged from adjusting the colors to make white text clearly visible to adjusting the palette to make it fit with a given visualization and the surrounding colors. The tasks were intended to familiarize the test subjects with the palette editing user interface, without explicitly explaining them how to change the colors. After the tasks were completed, the subjects were questioned about their experience with the palette editing user interface.

We noted that most subjects start by experimenting with the sliders. The result of this is that many subjects quickly learn how each slider affects the palette appearance. Several users noted afterwards that not all of the slider captions were clear to them, especially since the native language of none of the test subjects was English. However, the visual clues provided by each slider aid the users in learning its meaning. In some particular situations changing a slider influences the palette appearance very little or not at all; examples of this are the hue slider if saturation is low, the brightness slider if contrast is high, and the cold/warm slider if the hue is yellow. Some users encountered these situations early on, which increased the time needed to learn the workings of the user interface. However, most users were able to create the palettes they desired after a learning period of about 10 to 15 minutes.

In the second part of the user study, the aesthetical qualities of the generated sequential palettes were investigated. We acknowledge that aesthetics is just one aspect, and we are not sure if there is a relation with functional usability. Nevertheless, the experience at MagnaView is that for most customers aesthetics is important, and therefore this experiment was executed to provide us with more insight in this. For the experiment, a set of 48 palettes was formed by combining all sequential Brewer palettes of nine colors with 9-color generated palettes. The palettes were applied to a visualization in MagnaView and shuffled in three different orders to avoid ordering effects. Each participant was shown one of the three batches of 48 visualizations and had to rate each palette solely on its aesthetical properties, ranging from 1 to 5, from ugly to beautiful.

Two categories scored significantly higher than the others: the Brewer single-hue palette category (average rating

of 3.44), and the generated palettes having a  $w$ -value of 0.25 (3.40). We conclude that the generated palettes have a similar aesthetic quality to the hand-picked Brewer palettes. We also conclude that to generate an aesthetic palette, a  $w$ -value of around 0.25 is best. The palette that received the highest rating on average was a blue, generated palette with a score of 4.04. Furthermore, we evaluated the hue preferences of the participants by looking at the ratings of the single-hue palettes. The blue palettes scored highest with 3.75, followed by green and cyan. A more detailed description of the numbers and findings of this user study can be found elsewhere [Wij08].

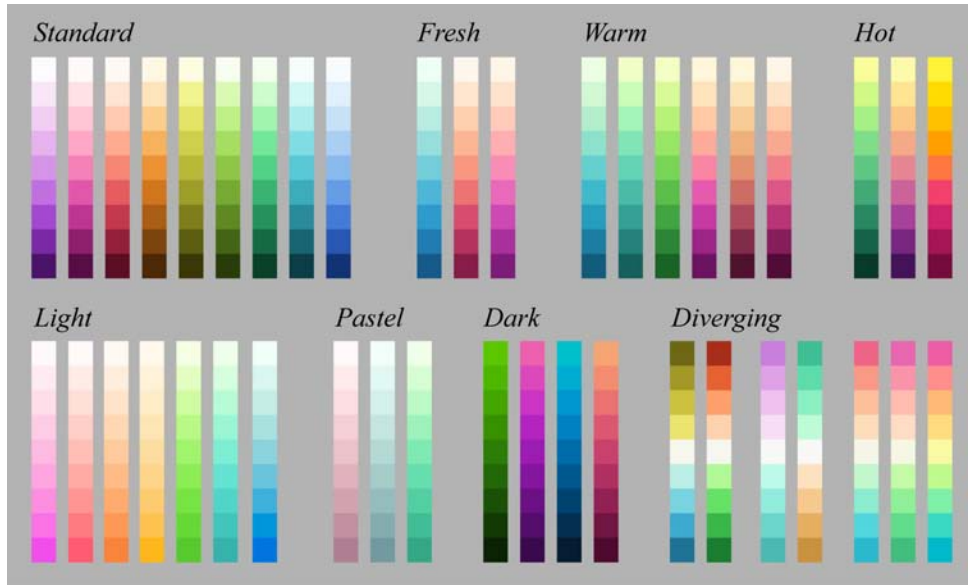
## 6. Conclusions

We have presented palette models that provide a flexible way of generating palettes for visualizations. By simplifying displayable CIELUV color space we are able to make better use of the irregularly shaped color space. We presented exact specifications of palettes instead of giving mere guidelines on palette generation. The palette appearance can be altered using high level intuitive parameters, which facilitates the palette design process.

We have paired our generated palettes to the Brewer palettes on Figure 9 to show that the generated palettes come close to the palettes designed by hand by an expert. Furthermore, our user study showed that palettes with a low  $w$ -value are rated equivalently to the highest rated Brewer palettes.

The multi-hue parameter makes it possible to generate a wide variety of multi-hue palettes. However, to make the lighter colors more saturated the lightness of the top point is decreased by a small amount, decreasing the used lightness range. This effect can be seen on the three pairs of sequential palettes on the right of Figure 9; the top colors are not as light as their Brewer counterparts. Further research on the multi-hue palettes might make it possible to create palettes with equal colorfulness, that utilize the complete lightness range. We did not find a simple solution to this problem.

Our palette definition makes it possible to define palettes in an intuitive and exact way. This definition can be used to automatically adapt palettes to fit certain situations. The pos-



**Figure 10:** Examples of generated palettes, taken from the set of preset palettes.

sibilities to use this definition to further automate the palette generation process is suggested for future work.

The PaletteView application for color space exploration can be obtained for research purposes from MagnaView B.V. Please contact [info@magnaview.nl](mailto:info@magnaview.nl).

## References

- [Bre99] BREWER C. A.: Color use guidelines for data representation. In *Proceedings of the Section on Statistical Graphics* (1999), pp. 55–60.
- [Bre05] BREWER C. A.: *Designing better maps. A Guide for GIS Users*. ESRI Press, 2005.
- [BRT95] BERGMAN L. D., ROGOWITZ B. E., TREINISH L. A.: A rule-based tool for assisting colormap selection. In *Proceedings of the IEEE Visualization 1995 Conference* (1995), pp. 118–125.
- [BT07] BORLAND D., TAYLOR R. M.: Rainbow color map (still) considered harmful. *Computer Graphics and Applications, IEEE* 27, 2 (2007), 14–17.
- [cie86] Colorimetry second edition. *CIE Publication No. 15.2* (1986).
- [Lev96] LEVKOWITZ H.: Perceptual steps along color scales. *International Journal of Imaging Systems and Technology* 7, 2 (1996), 97–101.
- [Lin] LINDBLOOM B. J.: [www.bruceindbloom.com](http://www.bruceindbloom.com).
- [Mag] <http://www.magnaview.com>.
- [MSK04] MEIER B. J., SPALTER A. M., KARELITZ D. B.: Interactive color palette tools. *IEEE Computer Graphics and Applications* 24, 3 (2004), 64–72.
- [Pha90] PHAM B.: Spline-based color sequences for univariate, bivariate and trivariate mapping. In *IEEE Visualization, Proceedings of the 1st conference on Visualization* (1990), pp. 202–208.
- [RO86] ROBERTSON P. K., O'CALLAGHAN J. F.: The generation of color sequences for univariate and bivariate mapping. *IEEE Computer Graphics and Applications* 6, 2 (1986), 24–32.
- [SKR99] SPENCE I., KUTLESA N., ROSE D. L.: Using color to code quantity in spatial displays. *Journal of Experimental Psychology: Applied* 5, 4 (1999), 393–412.
- [Tru81] TRUMBO B. E.: Theory for coloring bivariate statistical maps. *The American Statistician* 35, 4 (1981), 220–226.
- [VvWvdL06] VLIEGEN R., VAN WIJK J. J., VAN DER LINDEN E.-J.: Visualizing business data with generalized treemaps. *Transactions on Visualization and Computer Graphics* 12, 5 (2006), 789–796.
- [Wij08] WIJFFELAARS M.: *Synthesis of Color Palettes*. Master's thesis, Eindhoven University of Technology, dept. Math. and Comp. Science, the Netherlands, 2008.
- [ZH06] ZEILEIS A., HORNIK K.: Choosing color palettes for statistical graphics. *Research Report Series / Department of Statistics and Mathematics, Wien, Wirtschaftsuniv.*, 41 (2006).
- [ZM06] ZHANG H., MONTAG E. D.: Perceptual color scales for univariate and bivariate data display. In *Proceedings of the 30th International Congress on Imaging Science: ICIS '06* (2006), pp. 115–118.