Synthesis of color palettes

Wijffelaars, M.

Award date:
2008

TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computer Science

MASTER'S THESIS

Synthesis of Color Palettes

by

Martijn Wijffelaars

Supervisors:

Prof. dr. ir. Jack van Wijk (TU/e)
ir. Roel Vliegen (MagnaView B.V.)

Eindhoven, February 2008

Contents

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 thesis we focus on the use of color for data encoding, in which data values are mapped to colors. One solution is to use a continuous mapping, such that each value gets a unique color, which approach is especially popular in scientific visualization. An alternative approach, often used in cartography and information visualization is to use a discrete mapping. Here, a range of values is split into a number of sub-ranges, where each range is assigned a color. The resulting ordered set of colors is called a *palette*. Selecting the right combination of colors in such 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 peculiarities 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.

Therefore, the central question of this thesis is:

How to generate color palettes for visualization applications?

In this thesis, 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 generate a wide variety of palettes.

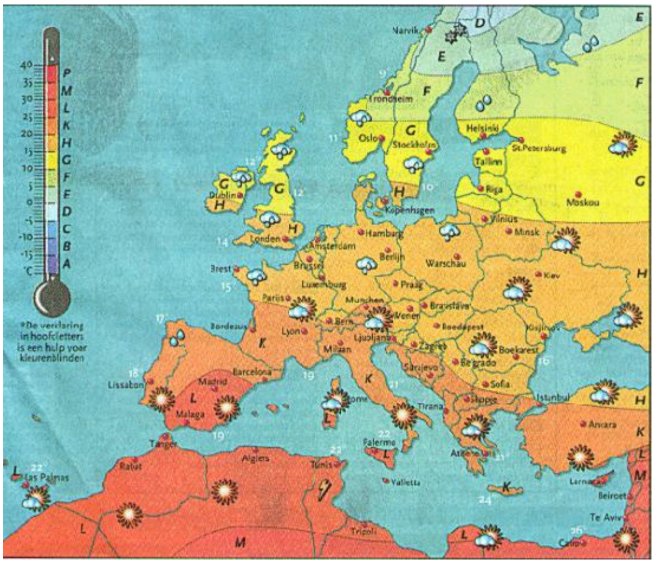
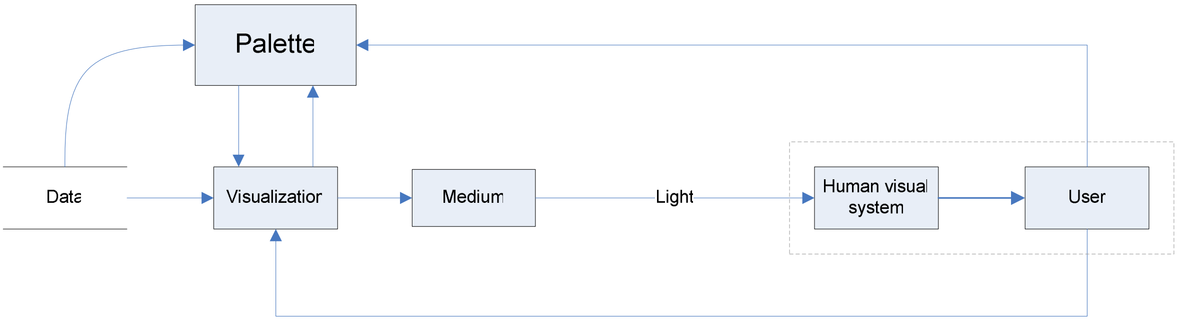


Fig. 1-1 A weather forecast visualization. The colors represent expected temperatures. The thermometer on the left shows the colors of the palette and their according temperature ranges. (Source: *Algemeen Dagblad*)

1.1 Objectives

The process of using palettes to color a visualization can be visualized by the diagram shown below. This diagram is used to get a structured view of the problem to be solved.



Essentially a **visualization** aids a **user** in analyzing certain **data**. The **palette** defines the colors used for certain aspects of the **visualization**. The colors are assigned using a mapping between **data** values and colors. The entities in the diagram are:

1. The **data** to be analyzed.
2. The **palette** is an ordered set of colors, which is created using information about the **data** and the **visualization**. The **palette** can be altered by the **user**, to better analyze the **data** or to fit his aesthetic preferences.
3. The **visualization** shows the **data**. The colors used in the **visualization** are determined by a mapping from **data** values to colors of the **palette**. The **user** can alter the **visualization**.
4. The **medium** transfers the rasterized image, generated by the **visualization,** to **light**. The **medium** can be, for example, a monitor, beamer or a printed image that reflects **light**.
5. The **human visual system** perceives the **light**.
6. The **user** interprets the signals from its **visual system**. Based on perception of the **visualization** the **user** can alter the **palette** or the **visualization**.

The path from visualization to the user is purposely split in several steps. It shows that, in the end, it is the human visual system that sends an interpretation of what it perceives to the user. Therefore, the set of colors, eventually outputted by the medium, should be tuned to the human perception.

The objective of this thesis is to present a method of generating palettes for computer applications. The requirements are:

- The palettes relay information in a perceptually correct fashion.
- The palettes fit several basic types of data (quantitative and qualitative).
- The palettes can be changed to fit different visualization-specific requirements and user preferences.
- The palettes can be altered in an intuitive way.
- The medium is constrained to be a CRT monitor, because it has a more predictable color representation than an LCD monitor, a beamer or a printer.

Finally, the palette generation techniques are integrated in the visualization tool MagnaView to show a practical application.

1.2 Overview

In chapter 2, we examine a small part of the human visual system to better understand how colors are perceived. In chapter 3, we present several color systems that can be used to specify colors on computer monitors. Chapter 4 gives an overview of the different types of palettes and the requirements needed to create them properly. Also, we give an overview of how others have designed and generated palettes. In chapters 5, 6 and 7, we present our models for palette generation of sequential, diverging and qualitative palettes respectively. In chapter 8, we give an overview of example palettes, we show how the palette generation techniques are integrated in the visualization application MagnaView, and we give a number of recommendations on how the palettes are best used in practice. Following that, chapter 9 presents our conclusions and a number of suggestions for future work. Finally, the appendix gives an overview of the used color transformation formulae and color distance functions.

The pictures used in this thesis are designed for use with CRT monitors. If this document is viewed on other media, the colors might differ substantially from the original pictures.

2 Color Vision

In this section we discuss a small part of the human visual system to form a basis for understanding human color vision perception. The pictures in this chapter originate from the www.handprint.com website of Bruce MacEvoy. The website contains a very extensive treatise on the subject of color vision, including information about human visual perception, and historical and modern color models.

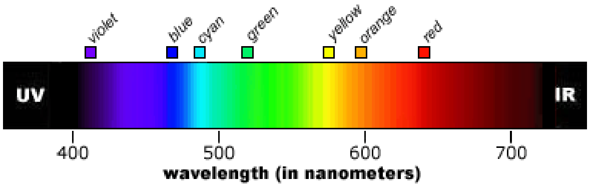


Fig. 2-1 The visible electromagnetic spectrum.

The sensation of color is created when light of a certain wavelength falls onto the human retina. The colors visible to the human eye consist of radiation in the range of wavelengths of about 370 to 730 nm (Fig. 2-1). A color can be characterized by its wavelength spectrum.

The light that falls onto the retina is processed by photoreceptors, better known as rods and cones. The rods respond to light of relatively low luminance levels, while the less sensitive cones are more responsive to higher luminance levels. There are three types of cones that together produce the sensation of color. Each cone has a different response curve, shown in Fig. 2-2. The curves represent how sensitive a cone is to different wavelengths of light. The cones are characterized by the peak value of the response curves: S, M and L, which respectively stand for short, middle and long wavelengths. The peak values are shown as white lines on the wavelength spectrum in Fig. 2-2. In the picture the curves are weighed by their population number on the retina. This is the reason that the S-cone curve is so low compared to the others; the number of S-cones on the retina is lower than the M- and L- cone numbers.

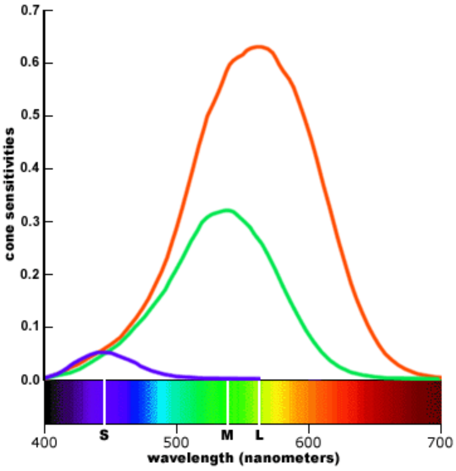


Fig. 2-2 The wavelength response curves of the S, M and L cones, weighted by their population.

A color of a certain wavelength distribution is converted by the cones to three cone response values (L, M, S). The cones essentially simplify each wavelength distribution to three values. Two important principles follow from this [27]:

- Trichromacy: All spectra can be reduced to precisely three values without loss of information with respect to the human visual system.
- Metamerism: All spectra that create the same trichromatic response are indistinguishable.

In other words, two colors will appear the same to us if they produce the same cone responses. All colors humans can perceive can thus be specified using only three values.

The peaks of the cone response curves lie on the wavelengths corresponding to blue, bluish green and yellowish green and not on blue, green and red as they are often depicted. Nevertheless, humans have a strong perception of blue, yellow, green and red. Opponent-colors theory explains why.

Late in the nineteenth century, Ewald Hering proposed his opponent-colors theory, which said that there are six elementary colors, and that these colors are perceptually arranged in opposing pairs: black/white, red/green and yellow/blue. One of the reasons for this observation was that people can describe colors in terms of combinations of these elementary colors, but they never name combinations of two opposing colors; there is no reddish green, or bluish yellow.

Modern opponent-colors theory backs up Hering's findings by providing it with a physiological basis. The cone response values are not directly transmitted to the brain, but combined to form three new signals:

- The luminance channel is based on the output of all cones.
- The red/green channel is based on output of the M cones subtracted from the L cones (L-M), shown in Fig. 2-4 (left).
- The yellow/blue channel is based on the summed output of the M and L cones minus the S cone output ((L+M)-S), shown in Fig. 2-4 (right).

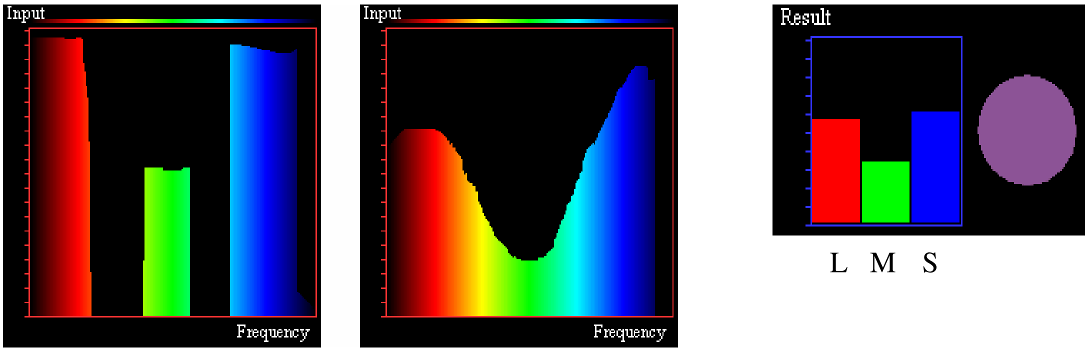


Fig. 2-3 Two spectrum distributions that, perceptually, yield the same color. The (L,M,S) stimulus is the same for both spectra and yield a purple color. Two spectrum distributions having the same cone responses are called *metamers*.

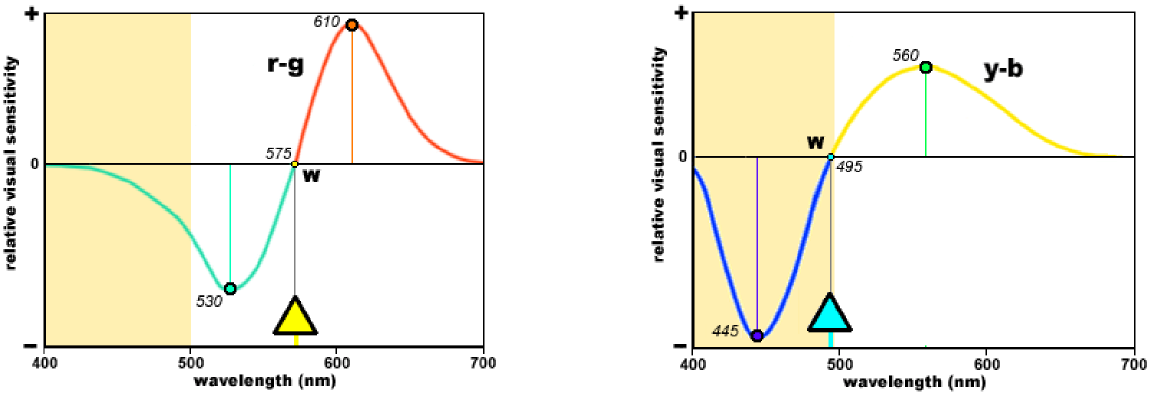


Fig. 2-4 On the left, the red/green curve resulting from subtracting the M curve from the L curve. On the right the yellow/blue resulting from the summed M and L curves minus the S curve.

These three signals are the basis for modern color systems discussed in chapter 3. Trichromacy relates to color vision at the photoreceptor level and opponent-colors theory relates to processing of the three cone responses. Many visual phenomena can be explained by trichromacy and opponent color theory.

An intuitive way of describing a color is to describe its *hue*, *lightness* and *saturation*:

Hue:

If people describe a color they usually name the hue. Red, blue, violet and lime green are all examples of hues. The hue is characterized by the dominant wavelength of a distribution spectrum.

Lightness:

Brightness, luminance, intensity and lightness are all closely related concepts.

- Brightness is the attribute of a visual sensation according to which an area appears to emit more or less light.
- Intensity refers to the measure over some interval of the electromagnetic spectrum of the flow of power [20].
- Luminance is radiant power weighted by a spectral sensitivity function that is characteristic of vision. So it is almost the same as intensity, except luminance is weighted by the human cone response curves. Luminance is often normalized according to a specified reference white.
- Lightness is the perceptual response to luminance. The relation between luminance and lightness is not linear, but roughly logarithmic [20].

In short, the most interesting term of the above is lightness, since we are interested in what is perceived.
The reason that we perceive blue colors to be darker is due to the fact that there are fewer S cones on our retina than M and L cones. The contribution of blue to our perception of lightness is therefore less than the wavelengths of other hues.

Saturation:

Colorfulness, chroma, saturation are also closely related [10]:

- Colorfulness is the attribute of a visual sensation according to which the perceived color of an area appears to be more or less chromatic.
- Chroma is the colorfulness of a stimulus relative to the brightness of a reference white.
- Saturation is the colorfulness of a stimulus relative to its own brightness.

However, the meaning of these terms may vary depending on the used color model. We prefer the term saturation as defined above.

Note that trichromacy is essential to the perception of saturation. If we only had two types of cones, we would not be able to determine the purity of the dominant wavelength. We would only be able to determine the lightness of a given distribution and a hue. However this hue would only be able to distinguish between two main colors. Furthermore, having only two cones would create far more metamers colors (Fig. 2-5).

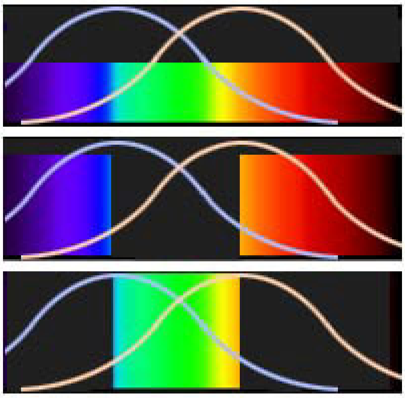


Fig. 2-5 Three spectrum distributions that are perceived as being the same color in a two cone system, i.e., they are metamers.

Apart from defining colors by lightness, saturation and hue, they are often defined by a name. However, these names are highly inaccurate representations of colors. According to research by Berlin and Kay in 1969 [31], there seem to be a number of terms that are consistently used across cultures. They defined a number of 'basic color terms', which are monoleximic, high-frequency and widely agreed upon by speakers of a single language. Monoleximic means that a word is not a combination of terms, so blue is monoleximic, but light blue or greenish blue are not. If a language uses only two color words, they are dark and light, or black and white. Languages with three color words use black and white plus red. The next color is either yellow or green, the sixth color is always blue and after that in varying order pink, purple, orange and gray. This is in line with opponent-colors theory, since the six most important color terms are black, white, red, green, yellow and blue according to Kay and Berlin.

Experiments by Post and Greene in 1968 on the naming of colors showed that of the 210 different colors supplied only 8 were consistently named. This is important to note, since Colin Ware suggests that colors with a distinct name are easier to remember [31]. Colors that are very close to ideal green or ideal red are easy to remember, in comparison to colors such as orange or lime green.

In the perception of colors two more phenomena play an important role: *simultaneous contrast* and *chromatic adaptation*. The former relates to the effect that colors appear different when viewed with different surrounding colors (Fig. 2-6). Chromatic adaptation refers to the adaptation of the eye to different viewing conditions: when the lighting of a scene changes, the perception of an object's color stays the same.

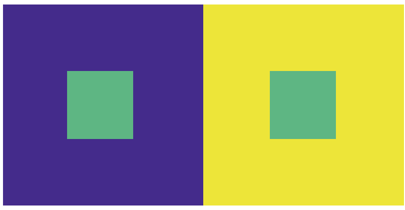


Fig. 2-6 Simultaneous contrast. The green square appears different when viewed with different surroundings.

3 Color Spaces

In this section, we give an overview of color spaces that can be used to specify colors. First, in section 3.1, we discuss absolute color systems, like RGB and HSV. In section 3.2, we discuss perceptually uniform color spaces, like CIELUV.

We use the convention HLS(270,60,20) to denote a color of a certain color space, in this case a HLS color with hue 270, lightness 60 and saturation 20. 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*.

3.1 Absolute color systems

The mostly used color system on computers is RGB, which stand for red, green and blue. The colors in RGB space are defined by three values that define the proportions of red, green and blue used to mix that color. These three colors are the three primary colors used in additive color mixing. RGB is a very direct way of specifying colors, since monitor and display hardware usually use pixels comprised of light emitting components of these three colors.

Often a RGB color is specified in hexadecimal numbers or in the range of $[0,1,..,255]^3$ due to the fact that underlying hardware uses bytes to represent the colors. We, however, prefer a more mathematical notation using the range of $[0..1]^3$.

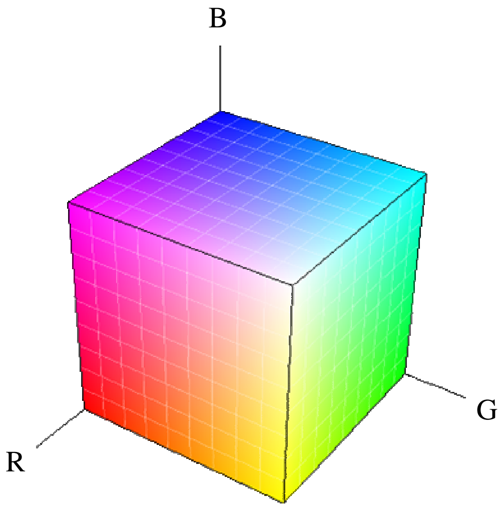


Fig. 3-1 The RGB cube

Since RGB values are comprised of three numbers, we can represent each color as a coordinate in three-dimensional space, using R, G and B as axes. All colors definable by RGB then make up a shape called the RGB cube, shown in Fig. 3-1.

The RGB space does not have intuitive dimensions; people do not think of a color as being 0.5 red, 1 green and 0.2 blue. A color system that does have intuitive dimensions is HSV, standing for hue, saturation, and value. HSV is also called HSB, where the B stands for brightness. HSV is a non-linear transformation of the RGB space. The transformation details of RGB to HSV can be found in the appendix. As with RGB, the HSV space can be represented using a three-dimensional model. We chose the commonly used cone to represent the HSV space, shown in Fig. 3-2a. The higher in the cone the lighter the colors get, the further from the center the more saturated, and the hue is determined by the angle. To show how the RGB cube is translated to this cone, the edges of the RGB cube are drawn in Fig. 3-2b. Three sides of the RGB cube make up the top cap of the cone and three sides make up the lower cone.

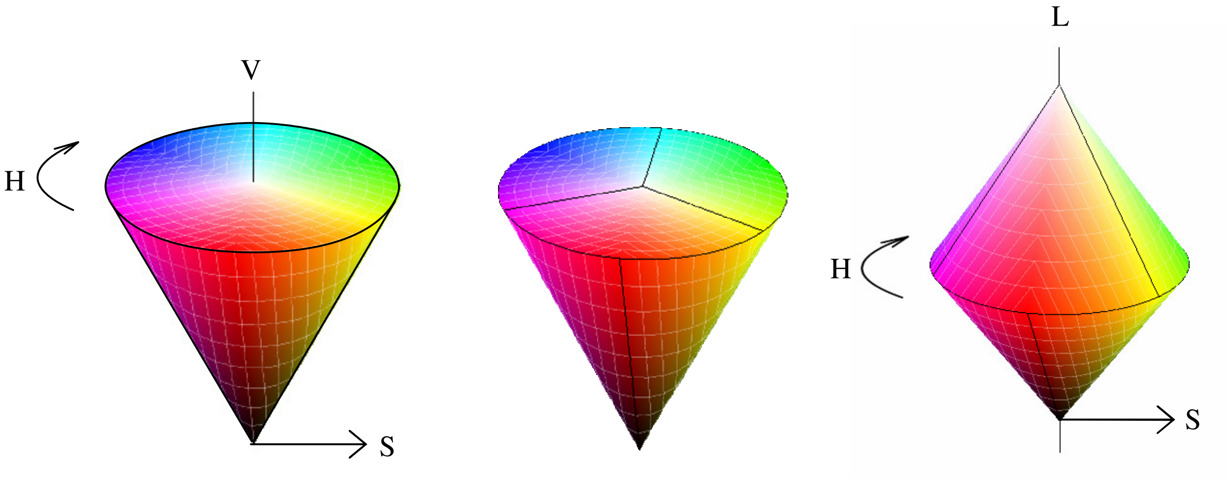


Fig. 3-2 a. HSV cone b. HSV cone with RGB edges c. HLS double-cone with RGB edges

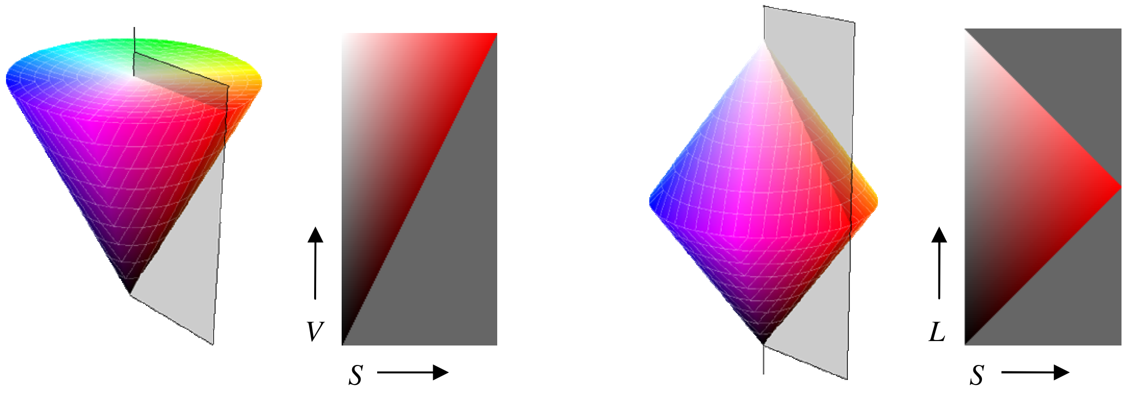


Fig. 3-3 a. HSV hue slice b. HLS hue slice

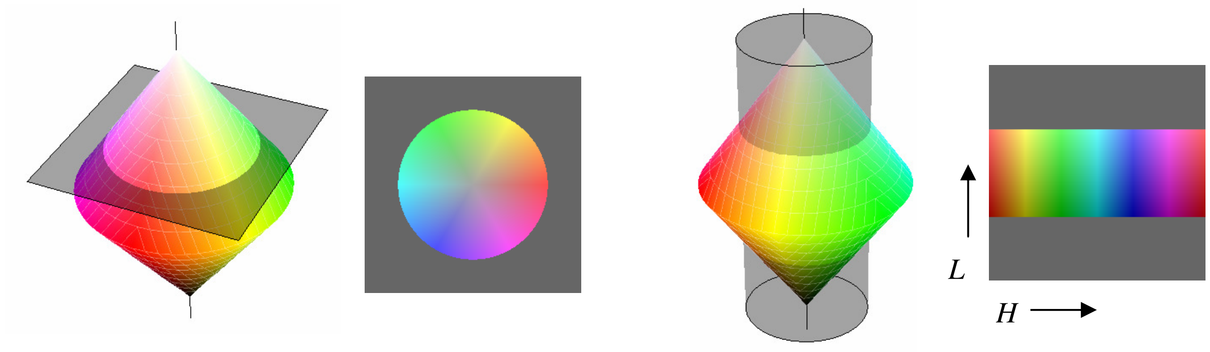


Fig. 3-4 a. A lightness slice of HLS space, the further from the center, the higher saturation. The angle represents hue.

b. A saturation slice of HLS space

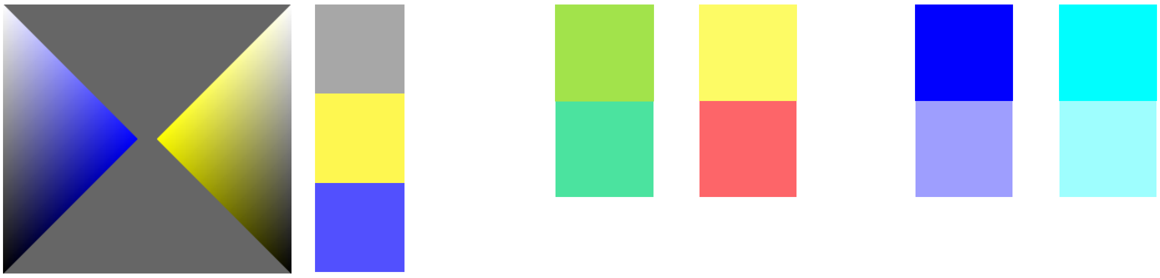
Another commonly used system is HLS, which stand for hue, lightness and saturation. Synonyms for this space are HSL and HSI, in which the I stands for intensity. The transformation of RGB to HLS is found in the appendix. The HLS transformation uses a slightly different calculation of the lightness and saturation. The HLS space is often visualized as a double-cone, shown in Fig. 3-2c. The most saturated colors in HLS have lightness of 0.5.

By taking slices of these three dimensional models more information can be obtained. In Fig. 3-3 hue slices of the HSV and HSL spaces are shown. A hue slice shows all colors of equal hue. The axes on the slice are lightness on the vertical axis and saturation on the horizontal axis. Fig. 3-4 shows a lightness slice and a saturation slice of the HLS space.

Although the HSV and HLS space have intuitive dimensions, they are not called *perceptually uniform*, which is defined as:

- Colors having equal values in a dimension are perceived to be equal in that dimension. For example, two colors of lightness 0.8 have the same perceived lightness.
- Equal steps in a certain dimension should yield equal perceptual differences in that dimension.

Fig. 3-5 shows three examples in which HLS is not perceptually uniform. The first picture shows two hue slices in HLS, which show that the most saturated colors of yellow and blue have the same L value in HLS, but not the same perceived lightness.



Hue slice of blue and yellow. The three colors on the right have equal lightness in HLS, but not equal perceived lightness.

Two sets of colors with equal hue steps. But not equal perceived hue steps.

Two sets of colors with equal lightness steps. But not equal perceived lightness steps.

Fig. 3-5 Examples of non perceptual uniformity in HLS.

3.2 Relative color systems

The color spaces defined in the previous section are called absolute color systems, meaning that they exactly specify all colors displayable by a monitor. The spaces in this section are relative color systems, which try to capture all visible colors, of which the monitor can display a subset. The subset of colors displayable by a certain display device is called its *gamut*. The systems presented in this section are attempts at perceptually uniform color systems.

The CIE, which stands for Commission Internationale de l'Eclairage, or in English International Commission on Illumination, is an organization founded in 1913 devoted to worldwide cooperation and the exchange of information on all matters relating to the science and art of light and lighting, colour and vision, and image technology [9]. In the early 1920s their attention was focused on predicting the perceptual response on a physical stimulus viewed in isolation by measuring the spectral power distribution of that stimulus. To find color matching functions that translate the spectral power distribution to tristimulus responses that match human perception, experiments were carried out in 1931 and revised in 1964. The subjects were asked to match a certain color, viewed on a neutral background, by adjusting three primary light sources of which the light was mixed. These lead to the specification of RGB color matching functions for a 'standard observer', representative of normal human color vision. These RGB color matching function were linearly translated into more convenient primaries, leading to the definition of XYZ. The function belonging to Y represents luminance, i.e., the total human cone response of a certain spectral power distribution.

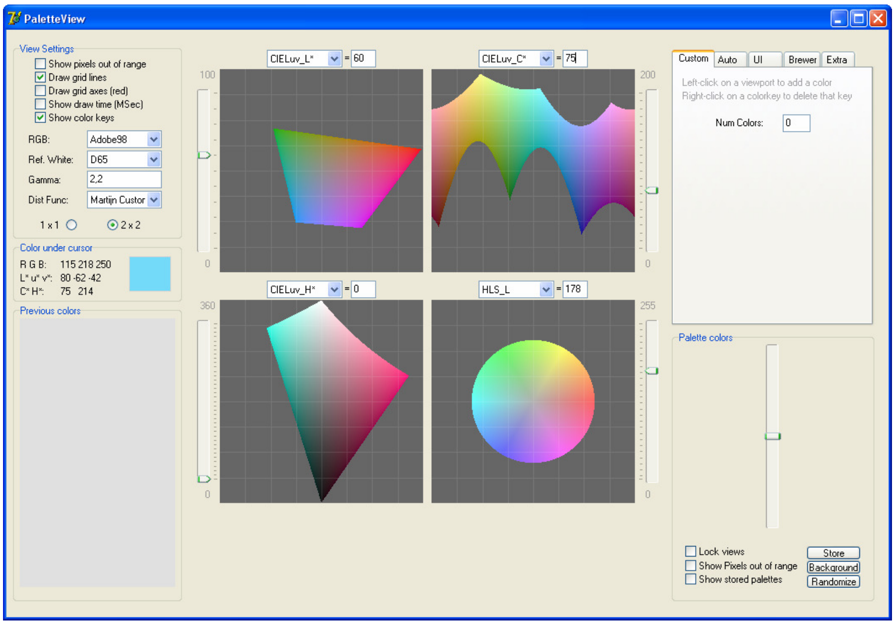
The XYZ values can be derived from the spectral power distribution of a stimulus measured by a spectrophotometer. These XYZ values can predict whether colors appear exactly the same to a human being, i.e., whether they are metamers. It can also predict whether a color appears lighter, duller or redder than another, but it can not predict by how much; XYZ can not predict perceived color differences. In order to shape XYZ into a perceptually uniform color system, several transformations of XYZ were proposed. In the 1960s about 13 to 20 of these formulae were used for different applications. In 1976 the CIE decided to end this diversity. Since no agreement was reached on which transformation to use as a standard, they proposed two, named CIELUV and CIELAB. These color systems were aimed at achieving perceptual uniformity. The CIELUV system is recommended in additive light source conditions, the CIELAB system for use under reflected light conditions [12][22]. Since we aim to create palettes on emissive displays, we choose the CIELUV space as a basis for the palette generation system.

CIELUV and CIELAB share the same lightness axis L^*, and the u^*, v^*, a^* and b^*, provide chromaticity information. Since the chromaticity axes do not have an intuitive meaning, the CIE proposed a transformation from CIELUV and CIELAB to respectively LCH_{uv} and LCH_{ab}, in which LCH stands for Luminance, Chroma and Hue. The complete formulae to transform RGB values to XYZ, to LUV to LCH_{uv} are described in the appendix. We only give the CIELUV formulae, since we use CIELUV for the generation of our color palettes. In the following, LCH is used to denote the LCH_{uv} space and u, v, a and b are used in stead of u^*, v^*, a^* and b^*.

In LCH space, colors can be specified that fall outside the monitor gamut, i.e., that can not be displayed on a monitor. We call the set of colors within the monitor gamut, the *displayable colors*. The RGB cube defines all displayable colors of a monitor. If an LCH color c yields an RGB color outside displayable color space, i.e., $\mathrm{RGB}(c) \notin [0..1]^3$, then the RGB value needs to be clamped back to that range to obtain a displayable color. The further a color is outside displayable color space, the worse the representation of the color in RGB. Therefore, our aim is to specify LCH colors that fall within the monitor gamut, to avoid irregularities resulting from clamping.

The HLS double-cone and slices through HLS give insight in how the HLS space is formed; we thought to find similar pictures of displayable CIELUV space. Even more so, since the formulae to calculate the RGB coordinates of a given LCH color are more complex than the HLS formulae. However, in books and on the internet we found few pictures of those. A figure that is often shown is the CIELUV space plotted within the xyY chromaticity diagram. However, this picture provides little useful information about the color space. Therefore, we decided to develop an application to give us more insight in displayable CIELUV color space. Furthermore, we hoped to get a better idea of how the parameters used to transform a color from RGB to LCH would influence the color space; these parameters being an RGB-matrix, a reference white and gamma value.

The PaletteView application (Fig. 3-6) can display slices of several color spaces. Such a slice is obtained by taking the colors on a plane intersecting with the displayable color space volume. In the following, we explain what is seen on several slices through CIELUV color space. We use these slices to draw conclusions about the palettes of others and about our own palettes.


PaletteView

View Settings
☐ Show pixels out of range
☑ Draw grid lines
☐ Draw grid axes (red)
☐ Show draw time (MSec)
☑ Show color keys

RGB: Adobe98
Ref. White: D65
Gamma: 2,2
Dist Func: Martijn Custor

1 x 1 ○ ● 2 x 2

Color under cursor
R G B: 115 218 250
L* u* v*: 80 -62 -42
C* H*: 75 214

Previous colors

CIELuv_L* = 60
CIELuv_C* = 75
CIELuv_H* = 0
HLS_L = 178

Custom | Auto | UI | Brewer | Extra

Left-click on a viewport to add a color
Right-click on a colorkey to delete that key

Num Colors: 0

Palette colors

☐ Lock views
☐ Show Pixels out of range
☐ Show stored palettes

Fig. 3-6 The PaletteView application shows four viewports that each displays a slice through a color space. On the top left an RGB-matrix, reference white and gamma can be selected.

The RGB cube contains all displayable colors of a monitor. Therefore, to visualize the volume of displayable colors in CIELUV, the RGB-cube is transformed to CIELUV space (Fig. 3-7). The lightness increases in the vertical L axis, the saturation increases for colors further from the center and the hue changes with the angle, just as with the double-cone of the HLS color space.

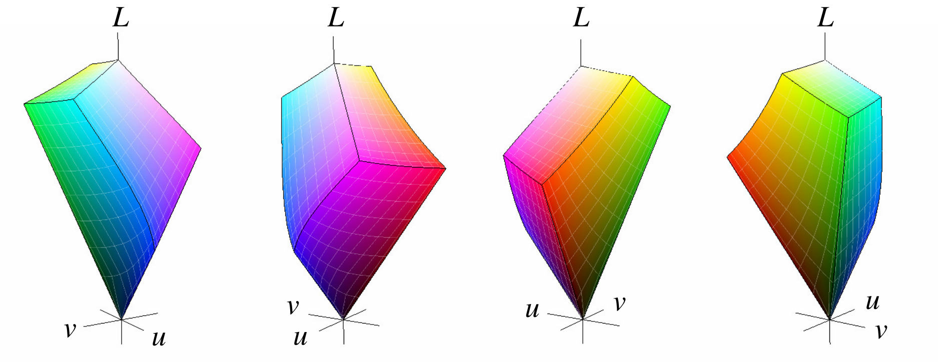


Fig. 3-7 The volume of displayable colors in CIELUV space, viewed from four sides.. It is obtained by transforming the RGB cube to CIELUV space.

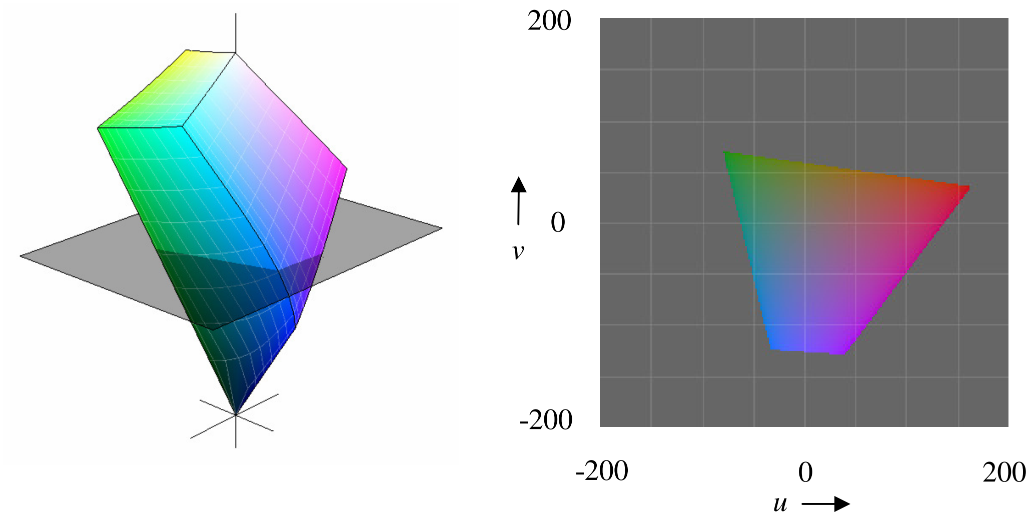


Fig. 3-8 A lightness slice of the CIELUV color space, with $L = 50$.

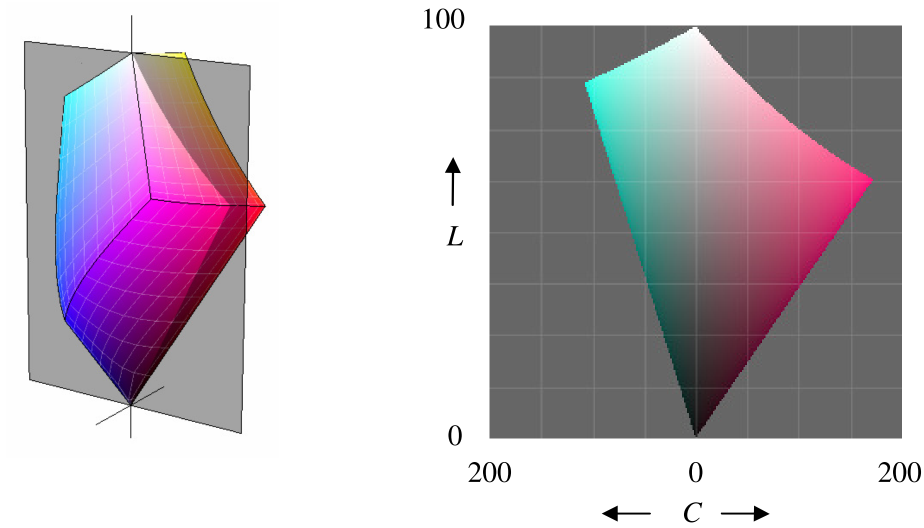


Fig. 3-9 A hue slice of the CIELUV color space, with $H = 0$ and $H = 180$.

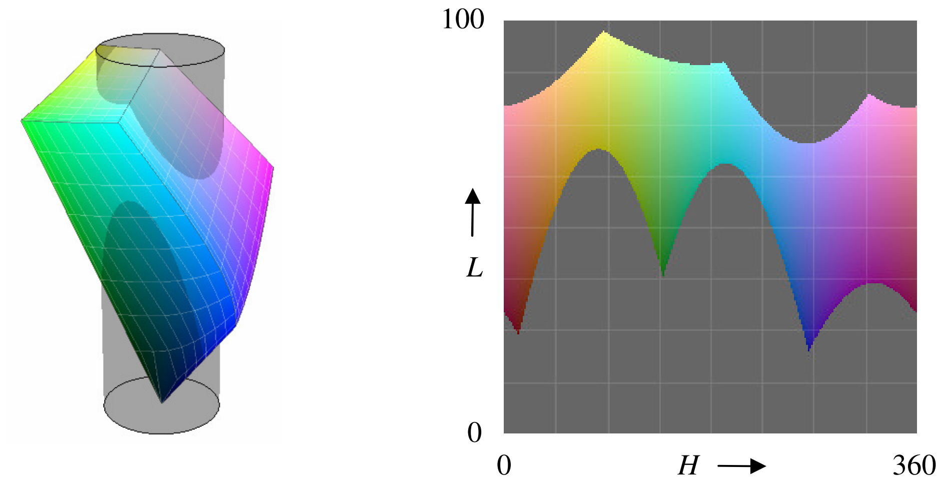


Fig. 3-10 A saturation slice of the CIELUV color space, with $C = 80$.

In Fig. 3-8, a slice of CIELUV is shown with colors of equal lightness. The *u*- and *v*-axes range from -200 to 200 since the highest saturation value of an RGB color, translated to CIELUV with the Adobe98 RGB-matrix, is about 200. Note that the color in the center of the slice, with LUV(50, 0, 0) is gray and colors further from the center are more saturated. In the PaletteView application it is possible to view slices of lightness values from 0 to 100, sliding the intersection plane up through color space.

In Fig. 3-9, a hue slice through color space is shown. The intersection plane with the RGB cube in CIELUV gives us a slice with two complementary hues. The lightness increases in the vertical direction and the further from the horizontal center the more saturated the colors are. These hue slices show us, for instance, that the most saturated color for hue 0 (red-pink) is darker than the most saturated color for hue 180 (cyan). To change the hues the intersection plane is rotated around the lightness axis.

The last slice we discuss is a saturation slice. Since LCH is a cylindrical transformation of the CIELUV space, the intersection surface yielding colors of equal saturation is not a plane but a cylinder, shown in Fig. 3-10. The slice shows all colors of saturation 80. The lightness increases in the vertical direction and the hues range from 0 to 360 in the horizontal direction. The slice shows that the lightest color with saturation 80 is yellow and the darkest is blue.

The PaletteView application can not only be used for exploring color space, it can also be used to analyze palettes. By clicking on the slices a set of colors can be selected. The colors are displayed on the slices by colored boxes. Each color can be dragged around on a chosen slice, and the position of the color on the other slices changes accordingly. This way, one can see how colors in a certain space relate to color in another space. It makes it possible to create and modify a palette, thus analyzing methods of palette generation. PaletteView also allows to load palettes created by others to analyze them within several color spaces.

4 Palettes

In this section we give an overview of various palette types. After that, we discuss the work of others, who have defined and generated palettes.

A palette is an ordered set of colors. In a visualization the colors of a palette are assigned according to a mapping from data values to colors. In case of univariate data, each value is mapped to a single color, with multivariate data, each combination of values is mapped to a single color. In this thesis, we only concern ourselves with generating a palette, without considering the mapping from data values to colors.

4.1 Discrete and continuous palettes

A discrete palette contains a finite number of colors; a continuous palette contains an infinite number of colors (Fig. 4-1).

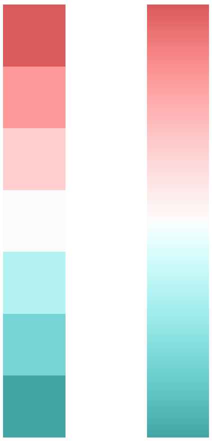


Fig. 4-1 A discrete and a continuous palette

In cartography often discrete palettes are used, while in data visualization people tend to use continuous palettes more often.

However, the choice of using a discrete or a continuous palette often depends on the type of input data. We use the following classification of data types defined by S. S. Stevens [31]:

- *Nominal*: These are data that display no ordering at all. Examples of nominal data values are: table, chair and couch. There is no logical ordering in this sequence.
- *Ordinal*: These are data values that are in some way ordered, i.e., the data values can be put in a sequence from low to high. Examples of ordinal data values are: very bad, bad, average, good, very good.
- *Interval*: These are data in which one can define a distance between two data values, but the zero point is arbitrary. The days of the week are interval data values, because one can say: "Wednesday is two days later than Monday", but we can not say that Thursday is three times as big as Tuesday. To be able to express that, a zero point on the data range is needed.
- *Ratio*: These are data values for which a zero point can be defined. So data values that represent the height of a person are ratio data, since we can state that 1.5 meter is twice as high as 0.75 meter. The zero point is defined at 0 meter.

For nominal and ordinal data types discrete palettes must be used. For interval and ratio data one can choose to use discrete or continuous palettes. Sometimes the data dictates this, such as with days of the week; there are only seven values, so a discrete palette would be appropriate. However, if the data values are made up of departure times of a train, i.e. interval data, then both types of palettes can be chosen. For all data that can use a continuous palette, a discrete palette can also be used, then the data values need to be categorized in a finite number of ranges.

We make the distinction between *quantitative* and *qualitative* data values:

- *Quantitative* data values display an ordering: ordinal, interval and ratio data values.
- *Qualitative* data values display no ordering: nominal data values.

4.2 Sequential, Diverging and Qualitative palettes

Cynthia Brewer suggests three base categories of palettes: sequential, diverging and qualitative palettes. More complex palette types can be created by combining these base palettes [5].

4.2.1 Sequential palettes

Sequential palettes display an ordering in colors from low to high. They can be used for any quantitative data type.

The most important requirements on a sequential palette are

- The colors display a clear ordering.
- The perceived color distances are representative for the intended distances.
- The colors are perceived as equally important; no color stands out between the rest.

The first two requirements are equal to the order principle and separation principle as defined by Trumbo [28]. Brewer and Levkowitz add the third requirement [5][13].

The rainbow color map is a widely used as a sequential palette, however, there are a number of reasons why it is not deemed to be a good choice for a sequential palette [3]. There is no clear perceived ordering in the rainbow palette, and most of the times the hue progression is not perceptually monotonically increasing, which results in some of the data region differences being accented and others being disguised. The rainbow color map is often defined as a palette that only changes in hue, but there is actually a lot of lightness contrast which accounts for a great part of the perceived differences.

It is recommended to use the luminance channel to achieve a perceived ordering in sequential palettes [3][4][31], because the human visual system is much better in assessing lightness differences than for instance saturation differences.

Research by Spence and Kutlesa also shows that the lightness channel is best for conveying quantitative information [26]. Their research showed that people are fastest in judging quantitative differences when only lightness difference is used, e.g., using a grayscale. However, they also showed that if a lightness increase is combined with a linear saturation and hue increase, this improves the ability to detect to which cluster a certain color belongs, others confirm this [3][19][30]. But the speed of judgement is somewhat lower than for palettes that only increase in lightness. A popular palette that increases in lightness and in hue is the heated object scale, or black body radiation scale which ranges from black, to dark red, to orange, to light yellow (Fig. 4-3).

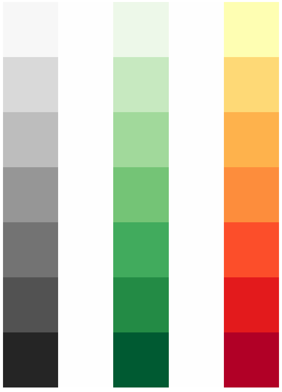


Fig. 4-2 Three seven color sequential palettes defined by Brewer.

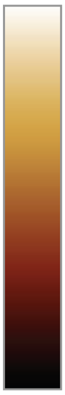


Fig. 4-3 Black body radiation scale [3]

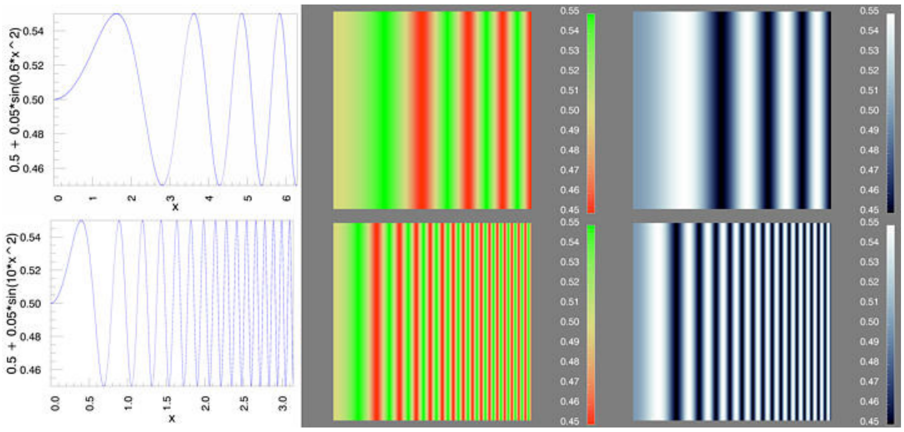
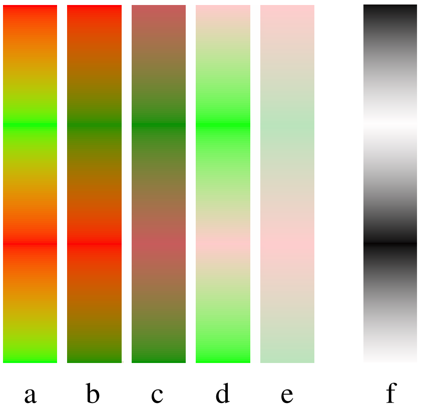


Fig. 4-4 Comparison of hue and lightness differences [24]. The top pictures show low frequency information, the bottom pictures high frequency information. However, since the hue difference pictures do have a high lightness difference, this picture is misleading.

It is said the human visual system is better in judging contrast differences when hue difference is used for low spatial frequency images, and when lightness difference is used for high spatial frequency images [10][24]. However, pictures used to show this are often hard to interpret correctly, as can be seen in Fig. 4-4 by Rogowitz and Treinish. In Fig. 4-5 the differences are stripped to bare hue differences without lightness and saturation contrast and show that hue difference alone does not yield a high contrast image. This is also due to the fact that hue differences yield more contrast at higher saturations. Hue differences might work better for low spatial frequency visualizations, but the lightness channel is far better in achieving perceivable contrast. This does not mean that hue contrast in combination with lightness contrast might work better than using only lightness contrast.



a: Red and green as used in Fig. 4-4.
b: Red and green at lightness of red.
c: Red and green at lightness of red and equal saturation.
d: Red and green at lightness of green.
e: Red and green at lightness of green and equal saturation.

f: White and black.

Fig. 4-5 According to CIELUV the original hue difference palette also displays a lightness differences. If this are removed and only a hue difference is left as on palette c and e, the contrast is very low.

The second requirement on sequential palettes is that the differences of colors are monotonically increasing in a way that equal distances in data values yield equal perceptual distances in the colors of the palette [13][26][31]. It is obvious that the more linearly the palette increases in lightness, the better the data values of colors of the palette are judged.

Finally, the third requirement states that all colors of a sequential palette are perceived as equally important. Although saturation differences alone do not yield a high contrast perception, it can be used to make a palette richer on information. Brewer suggests using higher saturation to accentuate more important categories [5]. This implies that saturation differences should be used with caution, because some categories might be regarded as being more important than others.

The previous leads us to believe that the ideal sequential palette is one that increases perceptually linear in lightness. The hue can be constant for faster judgement or can be varying for more accurate judgement. Variable saturation is allowed, but such that no color stands out between the rest.

4.2.2 Diverging palettes

Diverging palettes are used with quantitative data values. The difference with sequential palettes is that in a diverging palette a midpoint is defined on the data. This can either be the zero point or some other important point, such as the average of the data values.

The requirements on diverging palettes are similar to the sequential palette requirements, with the addition of:

- Colors on one side of the midpoint are clearly distinguishable from colors on the other side.
- If a neutral color is present, it should be clearly distinguishable from the rest of the colors

Sometimes, a clearly specified midpoint range is needed in a palette; this is represented by a neutral color. For example, if arrival times of a bus are represented by a palette, the range of acceptable times can be represented by the neutral color, while the arrival times that are either too early or too late are represented by different sides of the palette. When expressing profit and loss with a diverging palette, a neutral color might not be desired.

Brewer suggests using a light midpoint and dark end points [4]. For the single-hue palettes she uses a desaturated color as the neutral color, in the multi-hue palettes a light yellow color is used. That way, the neutral color is easy to detect since it is the lightest color in the palette.

One of the options that Rheingans, Robertson and O'Callaghan propose is a scale from one saturated color to gray to another saturated color [21][22], for instance a scale from blue, to gray, to yellow (Fig. 4-7). If the colors are chosen carefully, such a palette can have a monotonically increasing lightness. However, due to simultaneous contrast effects, it is hard to detect the gray color as a midpoint between the saturated colors.

The same arguments as for sequential palettes apply to each side of the diverging palette. Furthermore, the ordering is best expressed by using lightness contrast with a single hue. A varying hue can be used, but for diverging palettes this should be done with care. Using too much hue variation on one side of the palette might make it hard to distinguish between the two sides.

Brewer suggests using different hues for the two sides. Her palettes show that these hues need not necessarily be complementary. As long as the two hues are not too similar, a proper diverging palette is obtained.

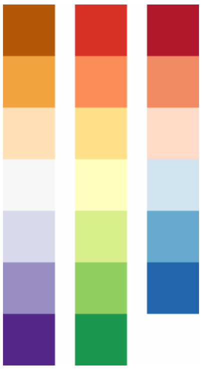


Fig. 4-6 Three diverging palettes defined by Brewer.

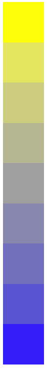


Fig. 4-7 A diverging palette with a lightness ordering.

4.2.3 Qualitative palettes

The qualitative palettes are used for qualitative data values, i.e., nominal data.

The requirements on qualitative palettes are:

- All colors are distinguishable.
- The palette displays no clear ordering.
- No color stands out between the orders. All colors are perceived as equally important.
- (Preferably colors are *nameable*.)

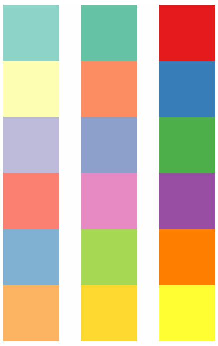


Fig. 4-8 Three qualitative palettes defined by Brewer.

Brewer suggests using colors of different hue to create qualitative palettes; in her palettes she allows the lightness and saturation of colors to change [4]. Ihaka, Zeileis and Hornik also vary the hue to create qualitative palettes. However, they use colors of equal lightness and saturation to ensure that no color stands out between the rest [12][32]. Healey uses a similar approach to find a maximum of distinguishable colors, only his approach allows saturation differences [11].

Furthermore, it needs to be noted that colors are easier remembered if they have a basic commonly accepted name [31]. Berlin and Kay define eleven colors that are consistently named across cultures: black, white, red, green, yellow, blue, brown, pink, purple, orange and gray. Post and Greene showed that there are some color names which are very precise, i.e., a similar name is given to only few color variations, such as yellow and red. Other colors are not so precise; a lot of colors are called blue. We call colors with a commonly accepted name *nameable*. Qualitative palettes with nameable colors are preferred, since their colors are easier to remember.

Note that although the sequential and diverging palettes can be discrete and continuous, the qualitative palette has to be discrete.

In conclusion, the qualitative palette should yield distinguishable colors by changing its hue. Whether lightness or saturation changes are allowed varies between approaches. Nameable colors are preferred.

4.2.4 Other palette types

Brewer states that other palette types can be derived from these basic palette types. An example of this is the bivariate palette. A bivariate palette is based on two variables instead of one. An example of this is shown in Fig. 4-9. Rheingans, Robertson and O' Callaghan also define a number of bivariate color palettes using various combinations of hue and lightness, mixing two sequential color scales [21]. Even a trivariate palette is suggested, in which three variables are mapped to the R, G and B primaries. Pham also defines a number of methods of generating bivariate and trivariate palettes [18]. However, most of these scales are hard to understand for an untrained user.

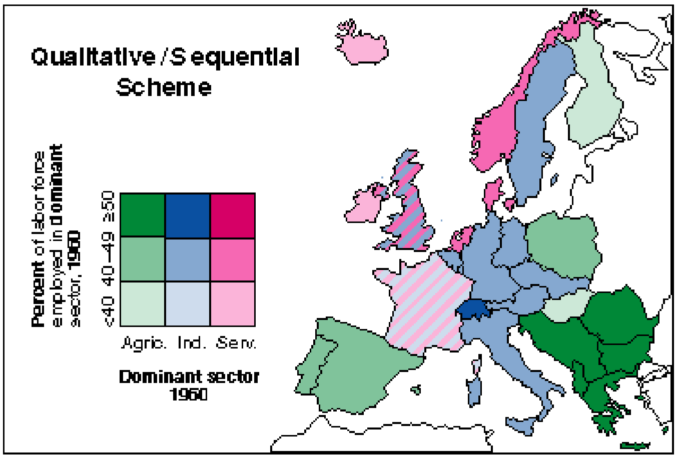

Qualitative / Sequential Scheme
Percent of labor force employed in dominant sector, 1960
<40 40–49 ≥50
Agric. Ind. Serv.
Dominant sector 1960

Fig. 4-9 An example a bivariate palette defined by Brewer [4].

Ware suggests to only use bivariate palettes in which one variable is mapped to hue differences and one to lightness differences [31], as shown in Fig. 4-9. These bivariate palettes can easily be derived by using several sequential palettes.

Other possible palettes are palettes in which one or several categories are accented by using very high saturation. Another possibility is to make a sequential palette in which the hue suddenly changes at a certain point. There are a lot of possible palettes that can be derived from the three standard palette types.

4.3 Related work

In this section we discuss the work of others who have defined palettes and proposed methods to generate them.

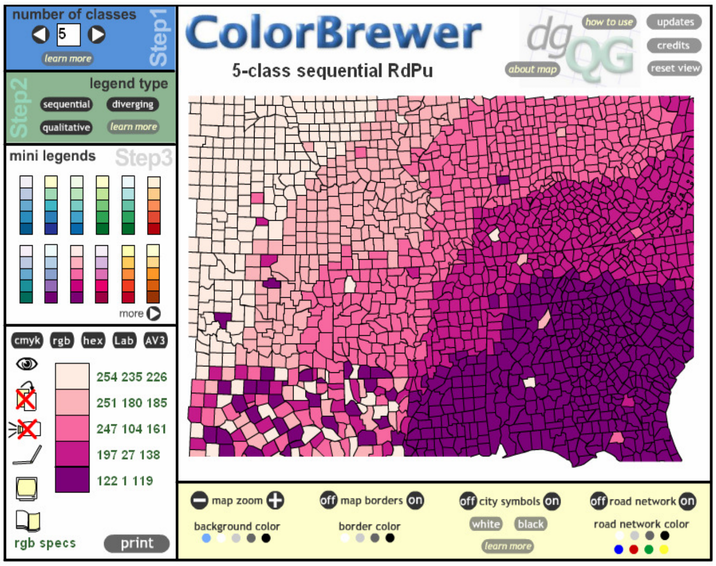
4.3.1 Color Brewer

Cynthia Brewer is a cartographer who defined a number of palettes based on her experiences as a graduate student and professor. On her website [6] these palettes can be applied to a map that displays low and high spatial frequency data. Moreover, each color is displayed completely surrounded by each other color, so simultaneous contrast effects can be judged. The website is a convenient tool for evaluating her palettes.

She has defined 18 sequential palettes, 9 diverging palettes and 8 qualitative palettes. The sequential and diverging palettes each have single- and multi-hue variants. All palettes are defined from 3 to at least 8 colors.

Furthermore, she defines a number of aspects of each palette: whether they can be viewed properly by colorblind people, whether they can be photocopied, and whether they can be used on LCD screens, projectors or on print.

We value her palettes highly, since they are not biased by a chosen color system or method of palette generation, but carefully created based on experience and expertise.



number of classes 5

5-class sequential RdPu

254 235 226
251 180 185
247 104 161
197 27 138
122 1 119

Fig. 4-10 A picture of the ColorBrewer website with which the Brewer palettes can be tested on a map with low and high spatial frequency data.

4.3.2 PRAVDAColor

PRAVDAColor is a color mapping module for Data Explorer that assists the user in choosing color palettes for visualizations [2]. It is based on PRAVDA, which stands for Perceptual Rule-Based Architecture for Visualizing Data Accurately. Rogowitz and Treinish developed a set of rules which used certain aspects of the input data to generate an appropriate color palette. These aspects include data type (nominal, interval, ordinal, ratio), minimum, maximum and whether there is a zero point in the used range of data values. But it also includes spatial frequency, which is used to determine whether a quantitative color palette gets a perceived ordering by using hue or lightness differences.

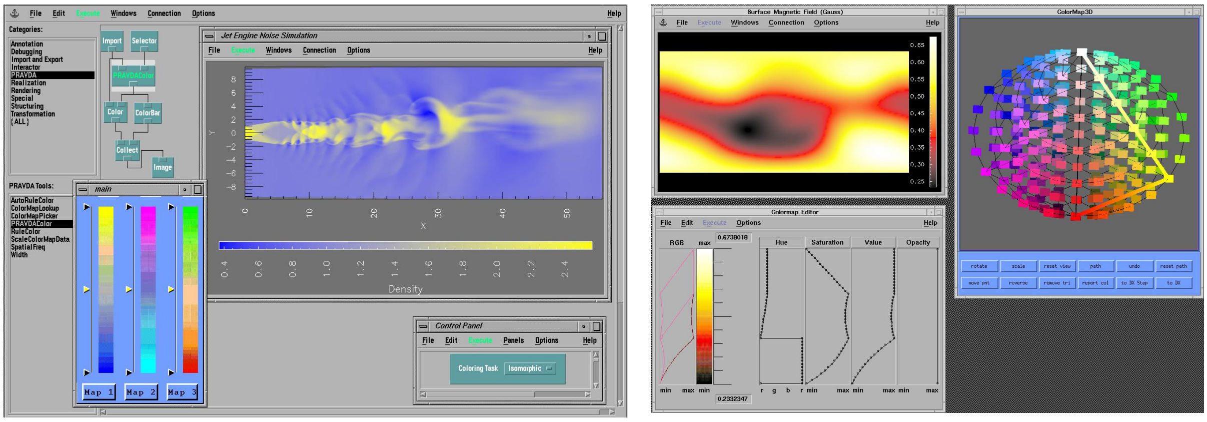


Fig. 4-11 Left: the Data Explorer using the PRAVDAColor module. Three preset palettes are suggested that are supposed to only differ in hue. Right: Color picking in PRAVDAColor happens by selecting color swatches from the HLS double cone.

They distinguish between two tasks: segmentation or highlighting. When segmenting, they again differentiate between high and low spatial frequency data: more steps can be used effectively for low spatial frequency data than for high [2]. In high spatial frequency data too many steps blur the visualization and make differentiation between colors impossible.

Based on these aspects, which are partially derived from the data and partially given by the user, a number of candidate palettes are suggested. The process of creating a palette is simplified for the user by giving preset palettes between which he can choose.

The HLS color keys of a palette can be edited by moving or selecting swatches from the HLS double cone displayed in Fig. 4-11. The palette is a result from linear interpolation between the color keys through HLS space.

This yields an excellent palette generation process: first analyzing the data, and possibly the user input, then supplying the user with preset palettes, which still can be altered to fit the user's needs. However, they do not use a *perceptually uniform* color system, which results in irregularities and non-monoticity in the color palettes.

4.3.3 Achim Zeileis, Kurt Hornik

Zeileis and Hornik provide a method of palette generation based on work of Ihaka [12][32]. They use the more perceptually uniform CIELUV color system to generate palettes. Although there are more methods using perceptually uniform color spaces, we discuss the method presented by Zeileis and Hornik seperately, because they provide clear example pictures of how their palettes are formed.

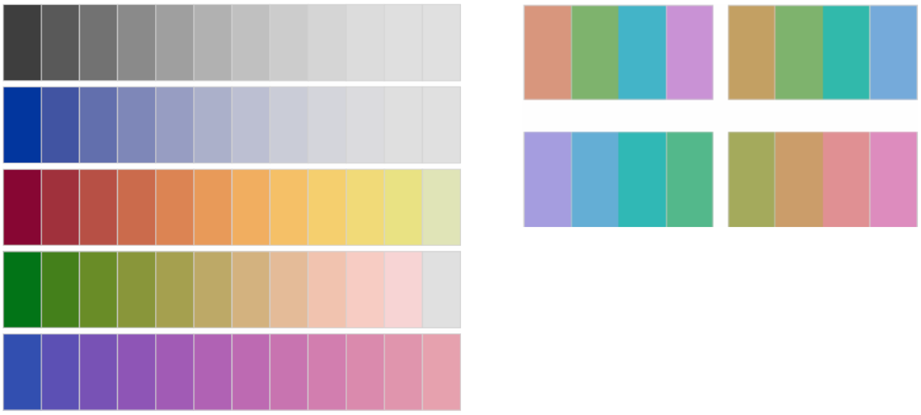


Fig. 4-12 On the left five sequential palettes are displayed that are generated with R by Zeileis and Hornik, on the right four qualitative palettes.

They differentiate between three base categories of color palettes that Brewer defined: sequential, diverging and qualitative. The palettes are specified in R, which is a language and environment for statistical computing and graphics. The sequential palettes are based on straight lines through color space. This means that it is never possible to generate a sequential palette with maximum lightness difference, unless it is a grayscale; the higher the lightness difference, the lower the saturation.

Furthermore, they are not able to cope with the boundaries of color space properly. This leads to an ad hoc method of generating palettes. In Fig. 4-13 two slices are shown with blue and green colors. They show a line through the blue colors that yields a nice sequential palette, however, the same line in terms of lightness and saturation can not be used for the green colors since it would include colors outside of displayable color space. Therefore they simply change the properties of the line, but fail to specify exactly what the line position should be.

The fact that they use a perceptually uniform color system like CIELUV to generate palettes is an improvement on the palette generation of PRAVDAColor. However, the use of straight lines limits the possibilities of palette generation. And because they do not explicitly define the color space boundaries, they always need to test whether a palette yields displayable colors.

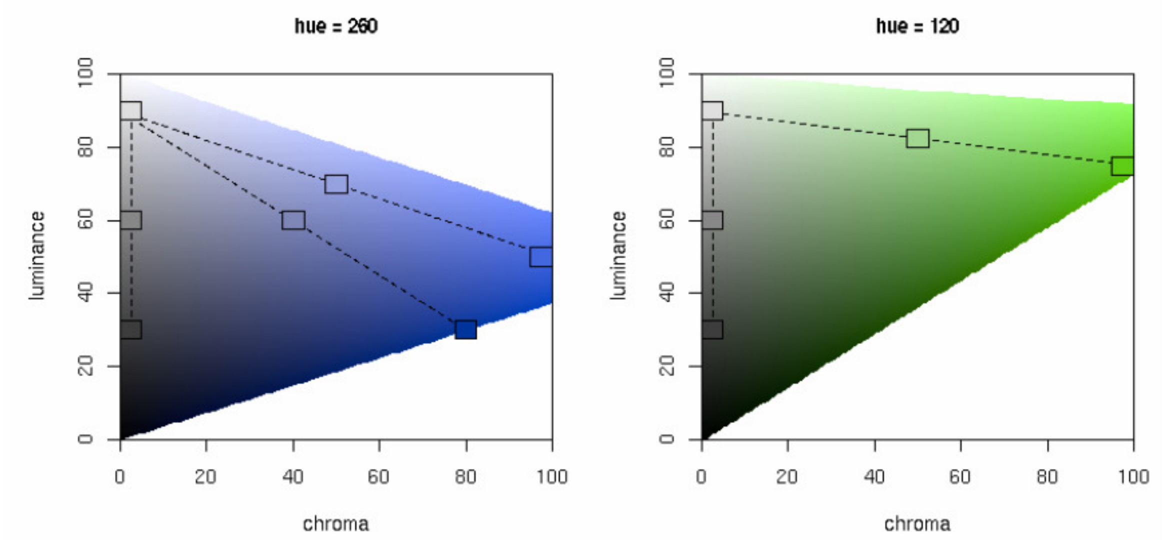


Fig. 4-13 Two intersections of CIELUV space each with colors of equal hue. The picture, by Zeileis and Hornik, shows the straight lines used to generate their palettes.

4.3.4 Other methods based on perceptually uniform color systems

Zeileis and Hornik are not the only ones who have generated palettes using perceptually uniform color systems.

Healey defined a method for generating qualitative palettes by using colors of equal lightness in CIELUV space [11]. Healey's approach is the same as the one of Zeileis and Hornik, except Healey did not require the colors to have equal saturation. He is therefore able to use a somewhat larger circle on a lightness plane. The center of the circle does not lie at a gray color, leading to colors of varying saturation and some hues being used more than others.

Levkowitz proposes a method for generating color palettes with equally spaced colors [13][14]. A path in RGB space needs to be chosen and then a set of colors along that path is converted to colors in a perceptually uniform colors space. By adding and removing colors the palette is 'linearized' according to a perceptually uniform color space to yield colors that are each at equal perceptual distance from each other. His method is independent of a color system, but in his papers he uses CIELUV.

Another range of palettes is created via a method originally created by Robertson and O'Callaghan [22]. Their univariate palettes are based on straight lines through a perceptually uniform color space, which could be either CIELUV or CIELAB. The straight lines need to take care that the colors are at equal perceptual distance from each other and yield a clear perceived order. However, by using straight lines, the irregularly shaped CIELUV color space can not be used to its full potential. For instance, the only way to achieve maximum lightness difference with a straight line is via gray scale.

Zhang and Montag follow the approach suggested by Robertson and O'Callaghan. They defined ten specific palettes based on line segments in CIELAB space and tested them with several visualizations [33]. Using line segments enables palettes to use a larger portion of color space, however, the discontinuities in first order derivative of the path, draw attention to those parts of the palette, thereby violating the equal-importance requirement on quantitative palettes.

Pham also uses the Robertson and O'Callaghan approach as a basis [18]. She uses B-splines, instead of straight lines, through perceptual color space in order to achieve higher flexibility in the palette definition. By using B-splines the perceptual color space can be used more effectively. The path is specified by giving several color keys and the proposed algorithm calculates the control polygon that defines the B-spline curve through the color keys. By using curves, the first order derivative of the path is continuous, generating a smooth palette.

4.4 Evaluation

Cynthia Brewer defined her palettes using experience and expertise; they are not based on a chosen color system or an algorithm for palette creation. Therefore, we use them as reference prototypes for our generated palettes.

Rogowitz and Treinish aid inexperienced users in creating a proper palette. They present a number of preset palettes, based on the data. A chosen preset palette can be changed by the user. However, they do not use a perceptually uniform color system.

Others do use perceptually uniform color system, but they fail to define concrete ways of generating palettes. A recent paper by Zhang and Montag said [33]:

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

This shows that no standard method of generating palettes with perceptually uniform color spaces is available. There are only guidelines on how palettes should be created. The techniques that use perceptually uniform color spaces make use of palettes based on color keys. This means that the user needs to have experience in generating palettes to be able to create a proper palette.

Instead of using color keys to specify palettes, we propose to use a limited set of parameters. The first three are hue, saturation and brightness, in accordance with the intuitive dimensions of color spaces. Further parameters are the number of colors of the palette, the cold/warm parameter that controls the hue range, and the contrast parameter that defines the overall lightness contrast of the palette. These parameters make it possible to define palettes that satisfy application-specific requirements; for instance, a palette in a certain lightness range ensures that white labels are always clearly visible, or, using a palette with lowly saturated colors makes it possible to highlight other aspects of the visualization with saturated colors.

In the following chapters, we present our palette generation techniques. In chapter 5, we propose a method for generating sequential palettes. In chapter 6, the model is extended to a diverging palette generation model. And finally, in chapter 7, a solution for generating qualitative palettes is presented.

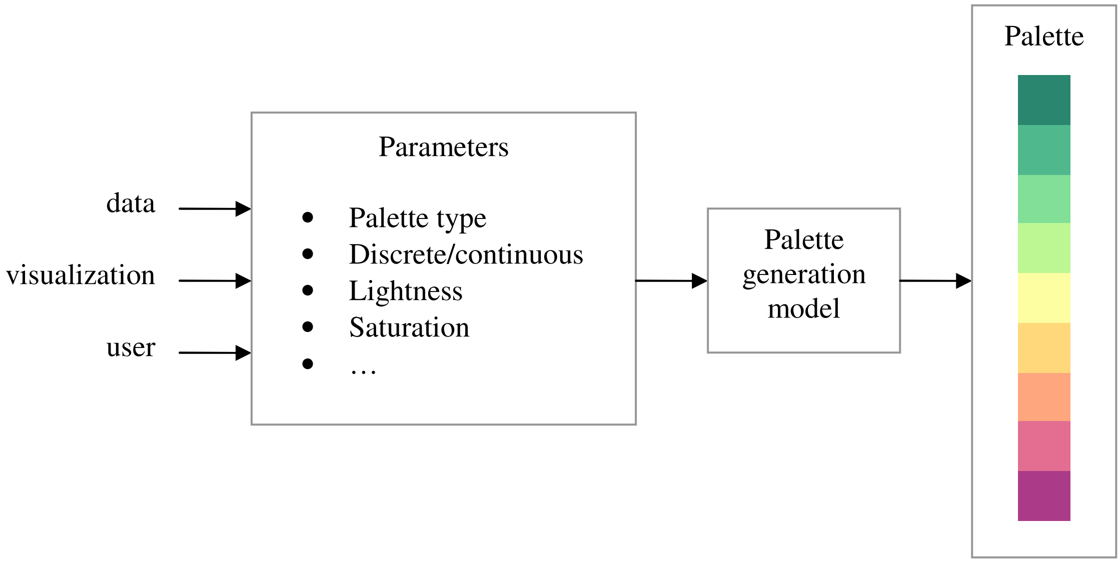


Fig. 4-14 Diagram of the intended method of palette generation. Using information about the data, the visualization and input from the user, a palette specification can be created. This is used to generate a palette.

5 Sequential palettes

In this section, we present a model for generating sequential palettes. Our aim is to generate both discrete and continuous palettes. The colors of a discrete palette can be defined as a subset of the infinite number of colors of a continuous palette. Therefore, we use a continuous palette as the basis for a discrete palette. This continuous palette is then sampled to obtain a discrete palette.

The first advantage of this approach is that one can easily switch from a continuous to a discrete representation of a palette. And the second advantage is that the number of colors of a discrete palette can be changed without having to generate a whole new palette; the continuous palette stays the same.

We used PaletteView to analyze the Brewer palettes within CIELUV color space. The RGB coordinates of the colors are converted to LCH coordinates. Fig. 5-1 shows the paths of three single-hue palettes of Brewer. Based on these 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 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 have smaller lightness distances between each other than the darker colors.
- The Brewer palette paths have saturation peaks at a different lightness levels, i.e., their shape is different for each hue. The peaks follow the shapes of the hue slices in CIELUV space, which also have saturation peaks at different lightness values.

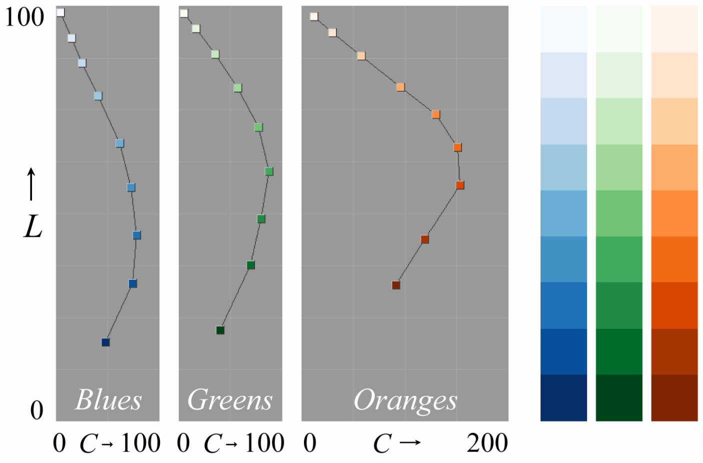


Fig. 5-1 The Blues, Greens and Oranges palettes of Brewer plotted in LCH space. The images are generated using PaletteView.

To create palettes similar to the Brewer palettes, we define a path through LCH space, on which colors are sampled. This path should have the following properties:

- The path itself is continuous.
- The path is smooth, i.e., the first order derivative of the path is continuous.

Although we require the path to be continuous, a non-continuous path can sometimes lead to useful palettes. An example of this is a palette in which the emphasis is put on two ranges of data by using different hues; the discontinuity is situated at the point where the hue changes. Another example is a palette in which one part receives emphasis because it is more saturated than the rest. Then, there is a discontinuity in saturation. Nevertheless, these palettes are not as common as palettes with a continuous path, and they can be constructed by using several continuous path palettes.

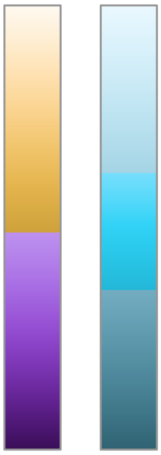


Fig. 5-2 Two sequential palettes. The left one with a hue discontinuity, the right one with saturation discontinuity

The second property the path should display is less trivial. If a path through color space does have discontinuities in the first order derivative, attention is drawn to that. This effect is similar to the Mach band effect. Mach bands can be seen at areas with abrupt changes in the first order derivative of a lightness profile, as seen in Fig. 5-3. We also observed these effects for abrupt changes in transitions of saturation or hue. Therefore, the path through color space is required to be continuous in its first order derivative.

To create a path through color space, we propose to use a curve, since that yields a smooth continuous path. For completeness and ease of reference, the complete definition of this curve is given below. Each part of the definition is explained in the sections that follow.

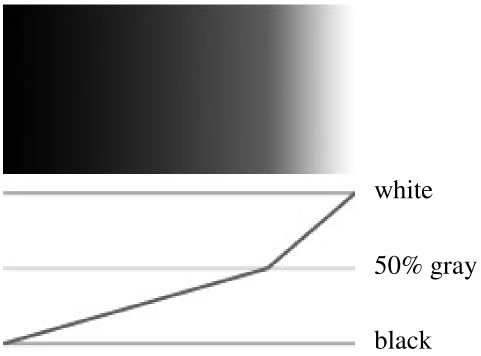


Fig. 5-3 Mach bands are a result of discontinuities in the first order derivative of the lightness profile. The Mach bands are visible at the point where the lightness is at 50% gray and at white.

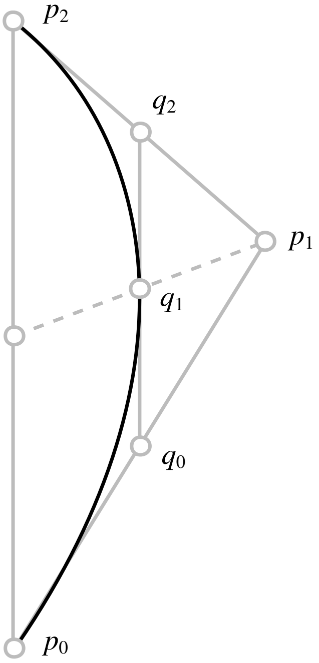
Single-hue sequential palette definition

Palette properties:

$\Sigma = [1..\infty]\times[0..360]\times[0..1]\times[0..1]\times[0..1]\times[0..1]$

$\sigma : \Sigma$

$\sigma = \langle N, h, w, c, s, b \rangle$



Palette:

$P_{seq} : \Sigma\times[0..1] \mapsto LCH$

$P_{seq}(\sigma, t) = C_{seq}(\sigma, (1-c)b + tc)$

Color curve:

$C_{seq} : \Sigma\times[0..1] \mapsto LCH$

$$C_{seq}(\sigma, t) = \begin{cases} B_0(2u), & u \le 0.5 \\ B_1(2(u-0.5)), & u > 0.5 \end{cases}$$

where $u = T(t)$ defined in section 5.4.

Bézier curve:

B_0 and B_1 are quadratic Bézier curves:

B_i: $[0..1] \mapsto LCH$, where i is 0 or 1.

$B_0(t) = B(\, p_0, q_0, q_1, t\,)$

$B_1(t) = B(\, q_1, q_2, p_2, t\,)$

B: $LCH\times LCH\times LCH\times[0..1] \mapsto LCH$

$B(b_0, b_1, b_2, t) = (1-t)^2 b_0 + 2(1-t)tb_1 + t^2 b_2$

Control points:

p_0 = LCH(0,0,h)

p_1 = MSC(h)

p_2 = LCH(100,0,h)

where MSC is defined in section 5.2.

$q_0 = (1-s)p_0 + sp_1$

$q_1 = \frac{1}{2}(q_0 + q_2)$

$q_2 = (1-s)p_2 + sp_1$

Default values:

$s = 0.6$

$b = 0.75$

$c = \min(0.88,\ 0.34 + 0.06N)$

First, the parameters defining the curve are grouped in six-tuple *σ*.

Palette properties:

$$\Sigma = [1..\infty]\times[0..360]\times[0..1]\times[0..1]\times[0..1]\times[0..1]$$

$$\sigma : \Sigma$$

$$\sigma = \langle N, h, w, c, s, b \rangle$$

The following one letter variable names are used to refer to the corresponding values of tuple *σ*:

- *N*: the *number of colors* of a palette. If $N = \infty$, the palette is a continuous palette, otherwise it is a discrete palette.
- *h*: the main *hue*.
- *w*: the multi-hue parameter that defines the *hue range* of the palette. If *w* = 0, all palette colors have hue *h*, otherwise a larger hue range is used. Until section 5.8, in which the multi-hue parameter is discussed, *w* is assumed to be 0.
- *c*: lightness *contrast* between the lightest and darkest colors of the palette.
- *s*: *saturation*.
- *b*: *brightness*.

P_{seq} represents the family of all sequential palettes that can be generated with the above definition. $P_{seq}(\sigma)$ represents a single sequential palette of which the properties are defined by *σ*, and $P_{seq}(\sigma, t)$ represents a single color of palette $P_{seq}(\sigma)$. We define $P_{seq}(\sigma, 0)$ to be the darkest color of a palette and $P_{seq}(\sigma, 1)$ to be the lightest color.

In the following sections we elaborate on all parts of the P_{seq} definition, every time restricting in more detail how the color curve is defined.

5.1 Control polygon surface

As mentioned earlier, we first define a method for generating single-hue palettes, which is extended to a multi-hue model. For a single-hue palette, the colors are restricted to a plane in CIELUV in which all colors have hue *h*. The surface needs to be restricted further, to minimize the proportion of colors outside displayable color space, in order to avoid color clamping.

The boundary lines of displayable color space on a hue slice are defined by intersections of surfaces:

The upper diagonal is defined by
$\{color \mid LCH(color).H = h \wedge RGB(color).X = 1\}$,
where *X* is *R*, *G* or *B*.

The lower diagonal is defined by
$\{color \mid LCH(color).H = h \wedge RGB(color).X = 0\}$,
where *X* is *R*, *G* or *B*.

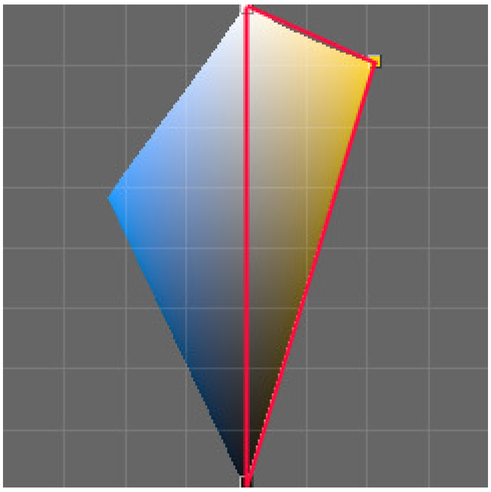


Fig. 5-4 A slice through CIELUV space with on the left colors with *H* = 245 and on the right *H* = 65. The red triangle approximates the boundaries of displayable color space for *H* = 65.

Although the shape of the color slice is different for all hues, the shape is always triangular, with a curved upper diagonal edge. This shape can be approximated to yield a simple-shaped control polygon for the curve.

The polygon is characterized by three points

p_0 = black,
p_1 = the most saturated color of *h,*
p_2 = white.

Although p_0 and p_2 are easily calculated, p_1 is more complex. This point is defined by the *most saturated color* of hue *h*, which we call the MSC(*h*). The calculation of MSC(*h*) is given in section 5.2. The three points defining the triangular surface in LCH terms are

p_0 = LCH(0,0,*h*),
p_1 = MSC(*h*),
p_2 = LCH(100,0,*h*)

LCH(0,0,*h*) is used, instead of LCH(0,0,0), in spite of the fact that they yield exactly the same RGB color. The reason for this is that these points are to be used as control points of a curve. By defining the color black like this, it is ensured that all intermediary colors are also of hue *h*. A similar argument holds for LCH(100,0,*h*).

Note that we deliberately use coordinates defined in LCH and not in CIELUV. The reason for this is that while interpolating between two colors, of which the hue is different, saturation is not kept equal. This can result in dull, unsaturated intermediary colors (Fig. 5-5). If LCH coordinates are used, the intermediate colors retain their saturation. For single-hue palettes this difference between color spaces is not crucial, since their hue is constant, but for multi-hue palettes it is.

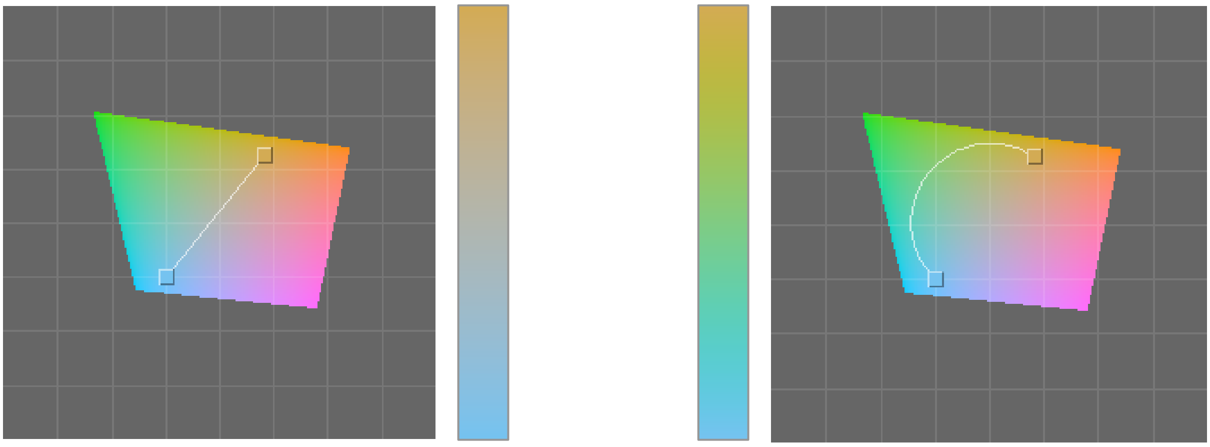


Fig. 5-5 Two lightness slices of CIELUV. Left, two colors are interpolated using CIELUV coordinates. Right, LCH coordinates are used.

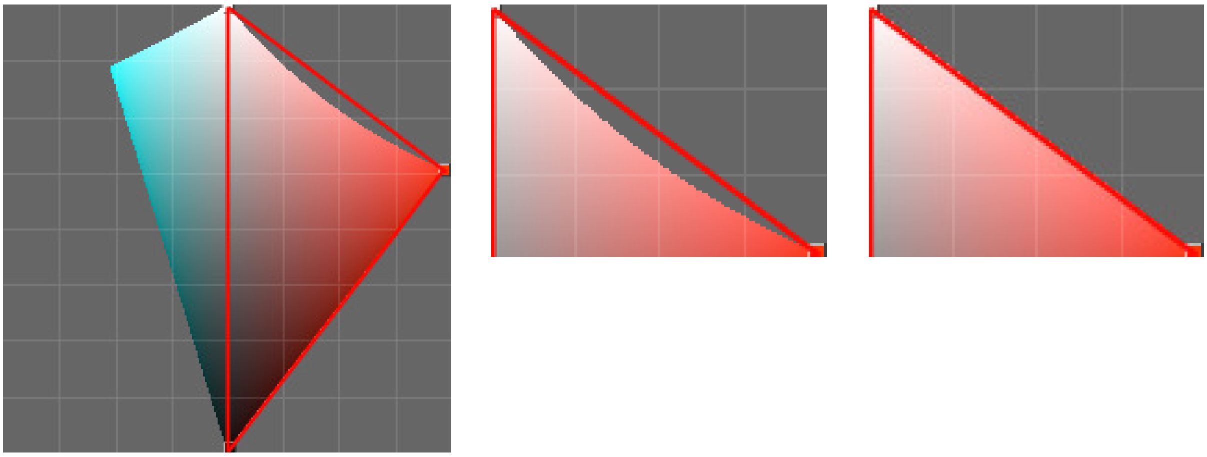


Fig. 5-6 Left, the approximation triangle for the red hue. Center, close up of colors within the triangle that are outside displayable color space. Right, the non displayable colors within the triangle are clamped.

Because the triangular surface is an approximation of the displayable colors surface, it does not exactly capture the displayable colors of all hues:

- There are colors within the triangle that fall outside displayable color space.
- There are colors within displayable color space that fall outside the triangle.

The first effect is most noticeable for reddish hues, an example of which is shown in Fig. 5-6. A part of the triangle does not have a displayable representation in RGB space. This, however, is easily solved by clamping the colors to the range $[0..1]^3$. This yields acceptable color representations as can be seen in Fig. 5-6.

The second effect is that some displayable colors fall outside of the triangle. This happens for colors of the hues in the range of 240 to 320, with the peak on hue 266 (pure blue), where the effect is greatest. This means, the triangular approximation surface does not contain the whole range of displayable colors. The colors that lie outside the triangle are the lighter, more saturated, bluish colors.

However, we discourage the use of very saturated palettes, hence this effect is of relatively little influence; it concerns only the very saturated colors of certain hues.

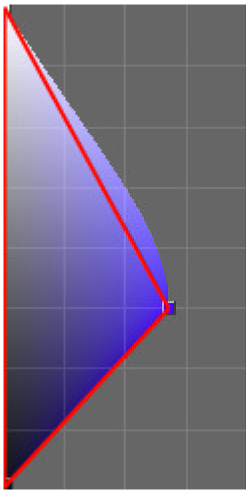


Fig. 5-7 A portion of displayable blue colors falls outside the approximation triangle.

5.2 Most saturated color calculation

This section focuses on the calculation of the MSC(h). The function takes an LCH hue h as its input and returns the LCH-color that is the most saturated displayable color for that particular hue h. The calculation of the MSC(h) is based on two properties:

1) The MSC(h) must lie on the LCH plane of h.
2a) The MSC(h) must be the most saturated color of that h.

The former property is easy to formalize, the latter one does not translate easily into a formula. We therefore reformulate the second property to be able to formalize it:

2b) The MSC(h) must be one of the colors on the edge of displayable color space of a hue slice in LCH space. We call the set of colors along this line the *color boundary*, CB(h), shown in Fig. 5-8.

$$\text{CB}(h) = \left\{ c \mid c \in LCH \wedge \left(\exists l : 0 \le l \le 100 : (\uparrow s : 0 \le s : c = LCH(l,s,h) \wedge RGB(c) \in [0..1]^3) \right) \right\}.$$

This equation says that $CB(h)$ is the set of colors, which are just displayable, but would not be displayable if we increased their LCH-saturation, i.e.,

$$\left(\forall h, c : c \in \text{CB}(h) : (\forall s : s > c.S : RGB\left(LCH(c.L,\, s,\, h) \right) \notin [0..1]^3) \right).$$

That means that the colors on the $CB(h)$ have at least one RGB-component at either 0 or 1. Because, if the saturation would be increased the color would just be outside RGB space, then one component would be less than 0 or greater than 1. For the colors of CB(h) holds

$$\left(\forall h, c : c \in RGB(\,\text{CB}(h)\,) : \begin{array}{l} c.R = 0 \ \vee \ c.G = 0 \ \vee \ c.B = 0 \vee \\ c.R = 1 \ \vee \ c.G = 1 \ \vee \ c.B = 1 \end{array} \right).$$

The colors on the edge of the rgb cube also have one of the RGB components at either 0 or 1. Therefore, the colors of the color boundary CB are all situated on the surface of the RGB cube. We now define MSC(h) in terms of $CB(h)$.

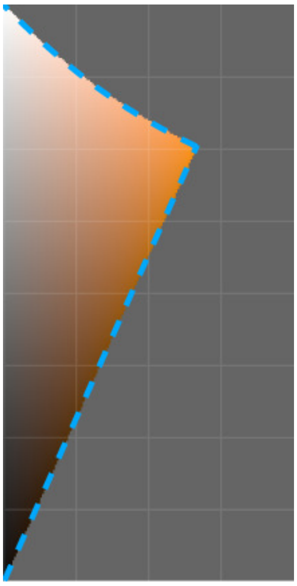


Fig. 5-8 The color boundary, CB(h), is depicted by the blue dashed line

$$\text{MSC}(h) = \left(\uparrow c.S : c \in CB(h) : c \right).$$

The MSC(h) is the color on the line CB(h) with the highest saturation. With MSC(h) defined in terms of CB(h) it is possible to define the path in RGB space on which the MSC colors lie. These lie on the edges of the RGB cube depicted in Fig. 5-9. The colors of the MSC lie somewhere on the surface of the RGB cube, because they are a subset of CB. If a MSC(h) would not lie on this path, then it must be moved over the surface of the RGB cube. It can only be moved in two ways, either closer to white, or closer to black. Both options lead to less saturated colors. Therefore, the most saturated colors must lie on this path.

The MSC(h) can now be determined by finding the intersection point of the path on the RGB-cube and the hue surface in LCH space. The path in RGB space is defined by 6 line segments. First, the line segment belonging to hue h is determined. This is done by calculating the LCH hues of the six vertices of the path on the RGB cube:

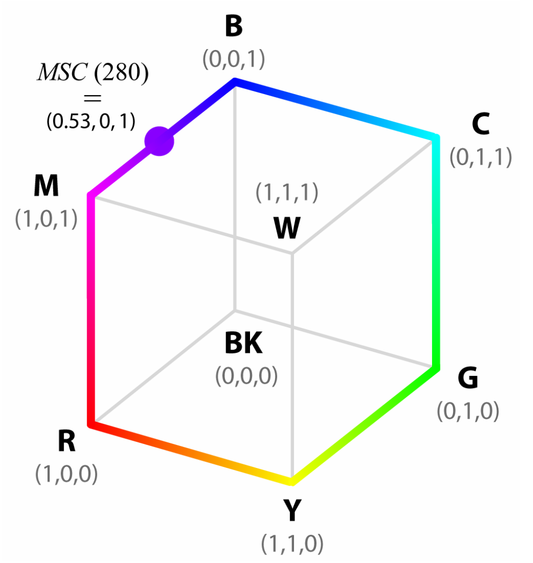


Fig. 5-9 The path of all MSCs, defined by six edges of the RGB cube. The MSC(280) is given as an example.

$h_0 =$	LCH(RGB(1,0,0)).H,	*red*
$h_1 =$	LCH(RGB(1,1,0)).H,	*yellow*
$h_2 =$	LCH(RGB(0,1,0)).H,	*green*
$h_3 =$	LCH(RGB(0,1,1)).H,	*cyan*
$h_4 =$	LCH(RGB(0,0,1)).H,	*blue*
$h_5 =$	LCH(RGB(1,0,1)).H,	*magenta*

Each line is defined by fixing one RGB-components at 0, one at 1, and one component is variable. Because we do not want to specify the calculation separately for each line segment, *c* is defined as an RGB color on the edge for which,

$c.\rho$ = the variable component
$c.\sigma$ = the component with value 0
$c.\tau$ = the component with value 1

Then for the six line segments ρ, σ and τ are defined as

Line segment	ρ (variable)	σ (min)	τ (max)
$h_0 \leq h < h_1$	G	B	R
$h_1 \leq h < h_2$	R	B	G
$h_2 \leq h < h_3$	B	R	G
$h_3 \leq h < h_4$	G	R	B
$h_4 \leq h < h_5$	R	G	B
$h_5 \leq h < h_0$	B	G	R

Note that there is one hue section where the upper bound is lower than the lower bound. For most RGB-matrices, this section will be $h_5 \leq h < h_0$, which for example could be $330 \leq h < 15$; then $h_5 \leq h \vee h < h_0$ is used.

With these definitions it is possible to calculate the intersection point in a generic way, without writing down the explicit calculations for each line segment.

For the first property of the MSC(*h*) the hue plane is defined in CIELUV space

$$\begin{aligned}\alpha &= -\sin(h)\\ \beta &= \cos(h)\\ \alpha u + \beta v &= 0.\end{aligned}$$

One further restriction will take care that the *u* and *v* are on the correct side of the L axis

$$-\alpha v \ge 0 \wedge \beta u \ge 0\,.$$

This requirement is always satisfied, because the resulting RGB color is restricted to be on one of the edges of the RGB cube. It takes care that the hue plane only intersects with the correct line segment and not the line segment at the complementing hue.

The MSC(*h*) is restricted by the CIELUV hue plane and the RGB path. The restriction on the RGB components can be translated to fit in the restriction for the CIELUV space. The formulae to convert RGB to CIELUV coordinates are given in the appendix. First the RGB restriction is written in terms of XYZ.

The definition of gamma correction is

$$c'.R = (c.R)^{\gamma},\ c'.G = (c.G)^{\gamma},\ c'.B = (c.B)^{\gamma},$$

which in terms of our general color components is

$$c'.\rho = (c.\rho)^{\gamma},\ c'.\sigma = (c.\sigma)^{\gamma},\ c'.\tau = (c.\tau)^{\gamma},$$

where *ρ*, *σ* and *τ* are dependent on *h* and defined by the table given above.

The definition of XYZ in terms of RGB

$$(X, Y, Z) = (c'.R, c'.G, c'.B)\cdot M$$

and with the matrix multiplication written out this is

$$\begin{aligned}X &= m_{RX}\cdot c'.R + m_{GX}\cdot c'.G + m_{BX}\cdot c'.B\\ Y &= m_{RY}\cdot c'.R + m_{GY}\cdot c'.G + m_{BY}\cdot c'.B\\ Z &= m_{RZ}\cdot c'.R + m_{GZ}\cdot c'.G + m_{BZ}\cdot c'.B\,,\end{aligned}$$

which in terms of our general color components, using the table above, can be written as

$$\begin{aligned}X &= m_{\rho X}\cdot c'.\rho + m_{\sigma X}\cdot c'.\sigma + m_{\tau X}\cdot c'.\tau\\ Y &= m_{\rho Y}\cdot c'.\rho + m_{\sigma Y}\cdot c'.\sigma + m_{\tau Y}\cdot c'.\tau\\ Z &= m_{\rho Z}\cdot c'.\rho + m_{\sigma Z}\cdot c'.\sigma + m_{\tau Z}\cdot c'.\tau\ \ .\end{aligned}$$

Now it is possible to simplify these equations by using the restrictions on the RGB color. We know that $(c'.\sigma)^{\gamma}=0$ and that $(c'.\tau)^{\gamma}=1$. Based on that the XYZ equation rewrites to

$$\begin{aligned} X &= m_{\rho X}\cdot c'.\rho + m_{\tau X} \\ Y &= m_{\rho Y}\cdot c'.\rho + m_{\tau Y} \\ Z &= m_{\rho Z}\cdot c'.\rho + m_{\tau Z} \ . \end{aligned}$$

The restriction on the CIELUV plane is rewritten in terms of X, Y and Z so the above definition can be used.

The restriction on the CIELUV hue plane

$$\alpha u + \beta v = 0$$

is rewritten by using the definition of *u* and *v*

$$\begin{aligned} u &= 13L(u'-u_n) \\ v &= 13L(v'-v_n) \end{aligned}$$

to

$$\alpha 13L(u'-u_n) + \beta 13L(v'-v_n) = 0 \ .$$

Note that u_n and v_n are constants denoting the *u* and *v* coordinates of the reference white. In the equation above, both terms have common factor 13*L*, which can be discarded because we can assume $L \neq 0$, since no hue is defined at $L = 0$. This leaves

$$\alpha(u'-u_n) + \beta(v'-v_n) = 0$$

$\equiv$

$$\alpha u' - \alpha u_n + \beta v' - \beta v_n = 0$$

$\equiv$

$$\alpha u' + \beta v' = \alpha u_n + \beta v_n$$

We continue with the left side of the equation and fill in u' and v' using these definitions

$$u' = \frac{4X}{X+15Y+3Z}$$

$$v' = \frac{9Y}{X+15Y+3Z}$$

which leads to

$$\alpha u' + \beta v'$$

$=$

$$\alpha\frac{4X}{X+15Y+3Z} + \beta\frac{9Y}{X+15Y+3Z}$$

$=$

$$\frac{4\alpha X + 9\beta Y}{X+15Y+3Z} \ .$$

Then *X*, *Y* and *Z* are substituted by their definitions in terms of $c'.\rho$

$$\frac{4\alpha(m_{\rho X}\cdot c'.\rho + m_{\tau X}) + 9\beta(m_{\rho Y}\cdot c'.\rho + m_{\tau Y})}{(m_{\rho X}c'.\rho + m_{\tau X}) + 15(m_{\rho Y}c'.\rho + m_{\tau Y}) + 3(m_{\rho Z}c'.\rho + m_{\tau Z})}$$

=

$$\frac{4\alpha m_{\rho X}\cdot c'.\rho + 4\alpha m_{\tau X} + 9\beta m_{\rho Y}\cdot c'.\rho + 9\beta m_{\tau Y}}{m_{\rho X}c'.\rho + 15m_{\rho Y}c'.\rho + 3m_{\rho Z}c'.\rho + m_{\tau X} + 15m_{\tau Y} + 3m_{\tau Z}}$$

=

$$\frac{(4\alpha m_{\rho X} + 9\beta m_{\rho Y})c'.\rho + (4\alpha m_{\tau X} + 9\beta m_{\tau Y})}{(m_{\rho X} + 15m_{\rho Y} + 3m_{\rho Z})c'.\rho + (m_{\tau X} + 15m_{\tau Y} + 3m_{\tau Z})} .$$

For readability the factors and constants are substituted by variables

$$\begin{aligned} f_1 &= (4\alpha m_{\rho X} + 9\beta m_{\rho Y}), \\ a_1 &= (4\alpha m_{\tau X} + 9\beta m_{\tau Y}), \\ f_2 &= (m_{\rho X} + 15m_{\rho Y} + 3m_{\rho Z}), \\ a_2 &= (m_{\tau X} + 15m_{\tau Y} + 3m_{\tau Z}). \end{aligned}$$

The division above can thus be rewritten as

$$\frac{f_1c'.\rho + a_1}{f_2c'.\rho + a_2} .$$

The equation of

$$\alpha u' + \beta v' = \alpha u_n + \beta v_n$$

can now be written to

$$\frac{f_1c'.\rho + a_1}{f_2c'.\rho + a_2} = \alpha u_n + \beta v_n .$$

This is rewritten to get $c'.\rho$ at one side of the equation

$$f_1c'.\rho + a_1 = (\alpha u_n + \beta v_n)(f_2c'.\rho + a_2)$$

≡

$$f_1c'.\rho + a_1 = (\alpha u_n + \beta v_n)f_2c'.\rho + (\alpha u_n + \beta v_n)a_2$$

≡

$$(f_1 - (\alpha u_n + \beta v_n)f_2)c'.\rho = (\alpha u_n + \beta v_n)a_2 - a_1$$

≡

$$c'.\rho = \frac{(\alpha u_n + \beta v_n)a_2 - a_1}{(f_1 - (\alpha u_n + \beta v_n)f_2)} .$$

The gamma correction is applied to get from $c'.\rho$ to $c.\rho$ by using the definition $c'.\rho = (c.\rho)^{\gamma}$

$$c.\rho = \left(\frac{(\alpha u_n + \beta v_n)a_2 - a_1}{(f_1 - (\alpha u_n + \beta v_n) f_2)} \right)^{\frac{1}{\gamma}}$$

and filling in the factors and constants leads to the final definition of $c.\rho$

$$c.\rho = \left(- \frac{(\alpha u_n + \beta v_n)(m_{\tau X} + 15 m_{\tau Y} + 3 m_{\tau Z}) - (4\alpha m_{\tau X} + 9\beta m_{\tau Y})}{(\alpha u_n + \beta v_n)(m_{\rho X} + 15 m_{\rho Y} + 3 m_{\rho Z}) - (4\alpha m_{\rho X} + 9\beta m_{\rho Y})} \right)^{\frac{1}{\gamma}} ,$$

where α and β are defined as

$$\begin{aligned} \alpha &= -\sin(h) \\ \beta &= \cos(h) \end{aligned} .$$

Because $c.\sigma = 0$, $c.\tau = 1$ and $c.\rho$ can be calculated, the definition of the MSC(h) in RGB space is complete. RGB color c can now be transformed to the desired color space.

5.3 Curve type

In section 5.1, the surface in which the curve must be contained is defined. In this section, we define the path the curve will follow on this surface.

The curve C_{seq} is based on two Bézier curves and the definition uses the previously defined points p_0, p_1 and p_2.

Definition of color curve C_{seq}

$$C_{seq}(\sigma, t) = \begin{cases} B_0(2t'), & t' \leq 0.5 \\ B_1(2(t' - 0.5)), & t' > 0.5 \end{cases}$$

$$\begin{aligned} q_0 &= (1-s)p_0 + sp_1 \\ q_1 &= \tfrac{1}{2}(q_0 + q_2) \\ q_2 &= (1-s)p_2 + sp_1 \end{aligned}$$

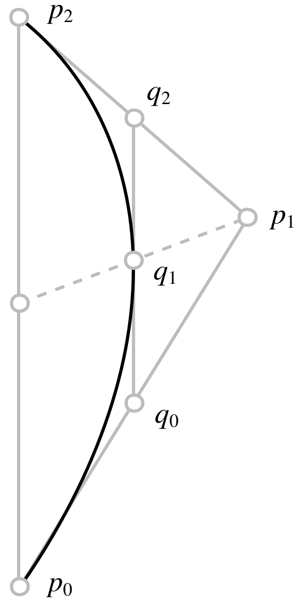
B_i: $[0..1] \mapsto LCH$, where i is 0 or 1.

$B_0(t) = B(p_0, q_0, q_1, t)$

$B_1(t) = B(q_1, q_2, p_2, t)$

$B: LCH \times LCH \times LCH \times [0..1] \mapsto LCH$

$B(b_0, b_1, b_2, t) = (1-t)^2 b_0 + 2(1-t)tb_1 + t^2 b_2$



The C_{seq} curve is needed to be able to vary the 'tension' of the curve. Fig. 5-10 shows a normal quadratic Bézier curve and its corresponding palette. If the saturation needs to be changed, the p_1 control point must be moved. However, then it is not guaranteed that the curve stays within the displayable color triangle.

Therefore, we define curve C_{seq}, which displays the following properties:

- The curve is continuous.
- The curve is continuous in its first order derivative.
- The curve is confined within the control polygon defined in the section 5.1.
- The curve has begin and endpoint interpolation, which makes it possible to use the complete lightness range for a palette.
- The curve has variable 'tension', which makes it possible to use the complete saturation range.

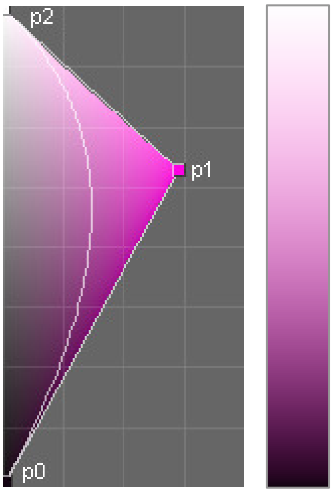


Fig. 5-10 A palette based on a single quadratic Bézier curve.

C_{seq} makes it possible to vary the saturation while keeping the curve confined within the convex hull defined by p_0, p_1, p_2. This is done by moving line q_0 q_2 proportional to parameter s, which represents saturation. If s = 0, then C_{seq} yields a totally unsaturated curve; if s = 1, then C_{seq} yields the most saturated colors possible (Fig. 5-11).

Curve C_{seq} satisfies the requirement that the curve is continuous in its first order derivative if $s \neq 1$. That is because the tangent of the begin or end point of a Bézier curve is parallel to respectively the first or last line of the control polygon. Therefore, because line q_0 q_1 is parallel to q_1 q_2 the tangents of B_0 and B_1 are equal at q_1.

If s = 1, the first order derivative has a discontinuity at q_1. The effect of this discontinuity is visible in Fig. 5-11 at the palette for which s = 1.

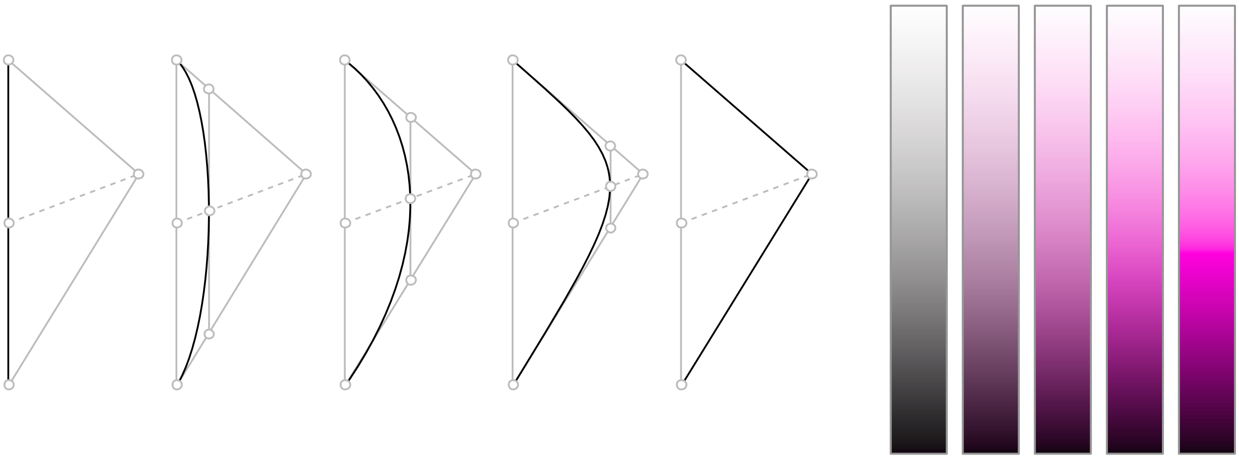


Fig. 5-11 Left: five C_{seq} curves with varying saturation: s = {0, 0.25, 0.5, 0.75, 1}.
Right: the five corresponding palettes.

5.4 Position on the curve

In this section, we discuss how to determine which t is needed to get the desired point from C_{seq}. We present a solution for normal Bézier curves, which can also be used for the compound curve C_{seq}.

For Bézier curves, point $B(t)$ is generally not the point at t percentage of the length of the curve. In other words, a Bézier curve is not arc length parameterized. An example of this is shown in Fig. 5-12a.

We define a function $T(\overline{s}):[0..1]\mapsto[0..1]$, which returns for a given normalized value $\overline{s}$ the corresponding value of t. This requires the inversion of the normalized arc length function $\overline{s}(t)$, which can not be done analytically. We therefore use a piecewise linear approximation of the normalized arc length $\overline{s}(t)$ with K line segments defined by the points

$$(t_i, \overline{s}_i)\,, \qquad i = 0, 1, .., K$$

$$\text{with } t_i = \frac{i}{K} \text{ and } \overline{s}_i = \frac{s(t_i)}{s(t_K)}\,,$$

where the arc length function $s(t_i)$ is defined as

$$s(t_i) = \sum_{j=0}^{i-1} D\big(B\left(j/K\right),\ B\left((j+1)/K\right)\big),$$

where $D(c_0, c_1)$ is the perceptual color distance defined in section 5.5.

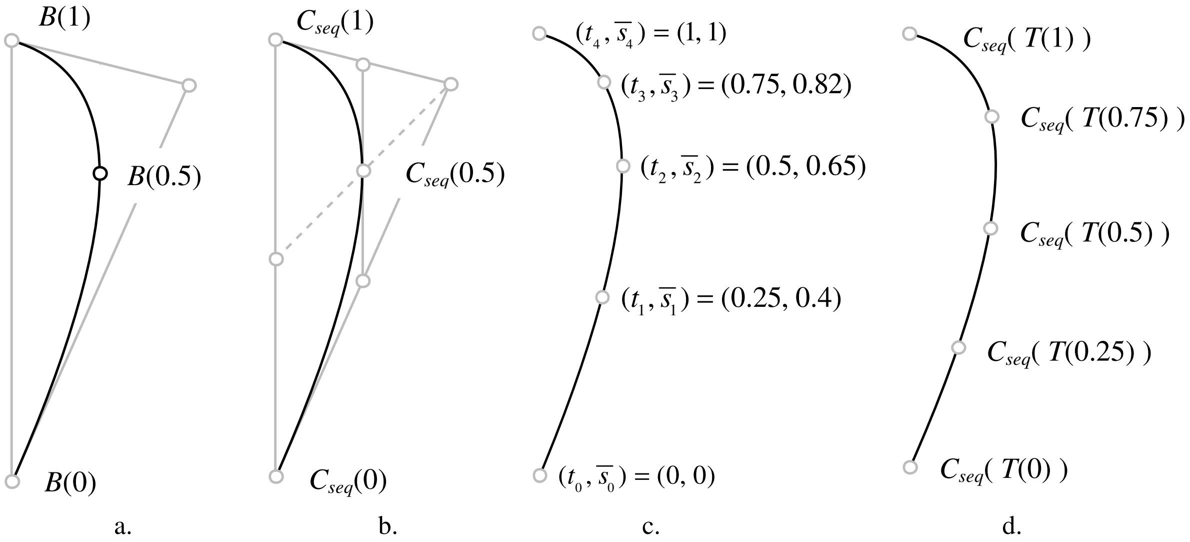


Fig. 5-12

a. A quadratic Bézier curve is not arc length parameterized; for example, $B(0.5)$ is not halfway the curve.

b. C_{seq} is not arc length parametrized, e.g., $C_{seq}(0.5)$ does not yield a point halfway on the curve.

c. Five points $(t_i, \overline{s}_i)$. These are used to approximate the value of $T(\overline{s})$.

d. Using T the desired points on the curve are obtained.

With these we can define $T(\overline{s})$ as

$$T(\overline{s}) = \sum_{i=0}^{N} w(K\overline{s} - i)\overline{s}_i \ ,$$

where

$$w(x) = \begin{cases} 1+x, & -1 \le x < 0 \\ 1-x, & 0 \le x \le 1 \ . \\ 0, & \mathit{otherwise} \end{cases}$$

The correct *t* for a Bézier curve *B* can thus be calculated using function *T*. However, C_{seq} is comprised of two Bézier curves. Fortunately, *T* can be adapted to fit any continuous curve. We adapt *T* by substituting *B* with C_{seq} and pose that as a solution to get the desired *t* for C_{seq}.

5.5 Distance function

In the previous section, function *T* makes use of a distance function $D(c_0, c_1)$, which denotes the perceptual distance between color c_0 and c_1. In this section, we define distance function *D*.

There are several distance functions proposed by the CIE. The CIELUV space was introduced in 1976 to yield a more uniform transformation of the XYZ space. CIELUV was intended to apply to comparisons of differences between object colors of the same size and shape, viewed in identical white to middle-grey surroundings, by an observer photopically adapted to a field of chromaticity not too different from that of average daylight [8]. With its definition, a method for calculating color distance was included

$$\Delta E^*_{uv} = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2} \ ,$$

which is the Euclidian distance between two CIELUV coordinates. The CIE also defined a hue difference [7]

$$\Delta H^*_{uv} = \sqrt{(\Delta E^*_{uv})^2 - (\Delta L^*)^2 - (\Delta C^*_{uv})^2} \ ,$$

$$\text{where } C^*_{uv} = \sqrt{(u*)^2 + (v*)^2} \ .$$

The CIE was aware that their definition needed some tweaking to fit different situations [8]:

> *"In different practical applications it may be necessary to use different weightings for ΔL*, ΔC* and ΔH*".*

And the distance function proposed at the time was not yet satisfactory, which is shown by a comment from the CIE in 1986 [8]:

> *"None of the existing color-difference formulae, including those recommended by the CIE, is entirely satisfactory and more experimental work is required to arrive at an accurate method of calculating color-differences."*

This shows that calculating the difference between two colors is a difficult task. It is even more complicated if colors are not viewed in ideal viewing conditions, because the perception of a color is heavily influenced by its surrounding colors and the lighting conditions.

The CIE revised their color distance formula in 1994 and defined it in terms of CIELAB colors. The CIE 94, as the new distance formula was called, is found in the appendix. This color difference function comes with a list of prerequisites which must be satisfied for CIE 94 to be used correctly, one of which is that the colors that are compared can not be more than 5 CIELAB units apart from each other. The list of prerequisites again shows that defining a uniform color distance formula is not a trivial task. The fact that the compared color samples have to look relatively alike to be compared stems from the fact that a lot of empirical data was used from textile industries. There, it is important to check whether a given sample does not differ too much from the original sample.

In 2001, the CIE again revised their color difference formula. The complete formula can be found in the appendix. The new formula was abbreviated as CIEDE2000. However it is significantly more complex than the CIE 94, in fact a paper was written to assist in implementing the formula since so much erroneous implementations of it exist [25].

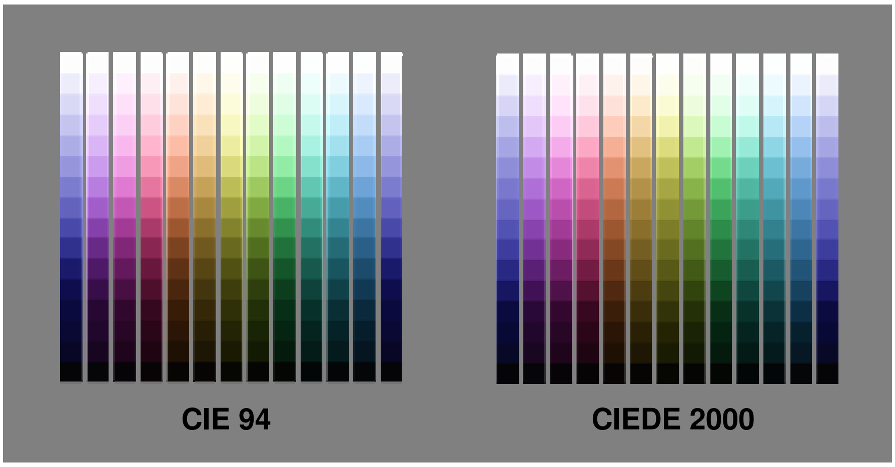


Fig. 5-13 Using the CIE94 or CIEDE2000 as distance functions yields palettes with colors that are not at equal perceptual distance to each other; the dark colors are too close to each other.

The palettes in Fig. 5-13 show that CIE 94 and CIEDE2000 are not suited to use for the sequential palettes generated with curve C_{seq}. It depends on the medium used to view this report, whether the palettes show this very clearly or not. The palettes were tested on CRT monitors and LCD displays and on both irregularities are clearly noticeable, the greatest of which were found in the darker colors of the palettes. The darker colors appear almost identical on most displays.

In Fig. 5-16 and Fig. 5-17, three palettes are plotted, which respectively use the CIE94 and CIEDE2000 distance function. The light and the dark colors are close to each other in CIELUV. With the Brewer palettes, the light colors are close too each other; the dark colors, however, are relatively far from each other (Fig. 5-14).

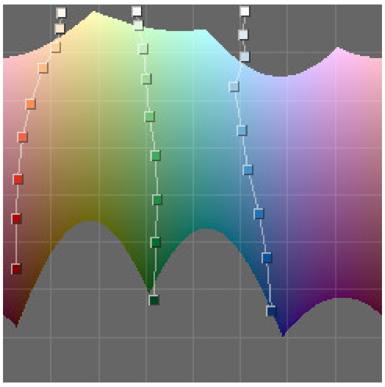


Fig. 5-14 Three Brewer palettes on a saturation slice; the light colors are relatively close to each other, while the dark colors are far from each other.

A known problem with the CIE 94 and CIEDE2000 is that they overestimate hue and saturation differences. A difference of 10 units in lightness yields a huge perceptual difference in comparison to a 10 unit difference in saturation [1]. On the website of Timo Autiokari very clear examples of this are shown. This underestimation of lightness differences is embedded in the CIELUV and CIELAB space, as shown by Fig. 5-15. CIE 94 and CIEDE2000 fail to compensate for this.

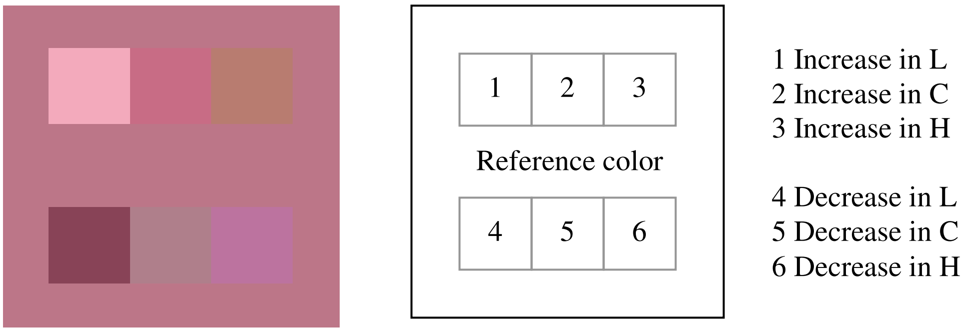


Fig. 5-15 A picture similar to the pictures of Timo Autiokari. It shows 6 samples that differ 20 units in LCH space with the reference color.

Therefore, we define a new distance function *D*, in order to make the distances of the colors on C_{seq} more like the Brewer palette distances: the darker colors are further apart and the lighter colors closer to each other. In other words, *D* needs to increase the color distances for low L-values and decrease it for high L-values. We used a natural logarithm to establish that. We obtained satisfying results using the following definition of *D*:

$$D(c_0, c_1) = \left| \ln\left(\frac{125 - c_1.L}{125 - c_0.L} \right) \right| .$$

The *D* function is based solely on the lightness components of colors, since that is the most important color attribute for perceiving an ordering in sequential palettes. This leads to several pleasant properties:

- $C_{seq}(\sigma, t)$ yields colors of similar L value for each *h*. This is not only convenient for sequential palettes; it is also useful when the sequential palette definition is extended to diverging palettes in section 6.
- Using only the L-value makes it is easier to define discrete palettes with colors at equal distance under *D* in section 5.7.

Fig. 5-18 shows the same palettes as in Fig. 5-16 and Fig. 5-17, only here the *D* function is used. The figures on the next page, show an overview of the palettes generated using CIE94, CIEDE2000 and *D*. The palettes generated using *D* yield colors that are more uniformly spaced (Fig. 5-20).

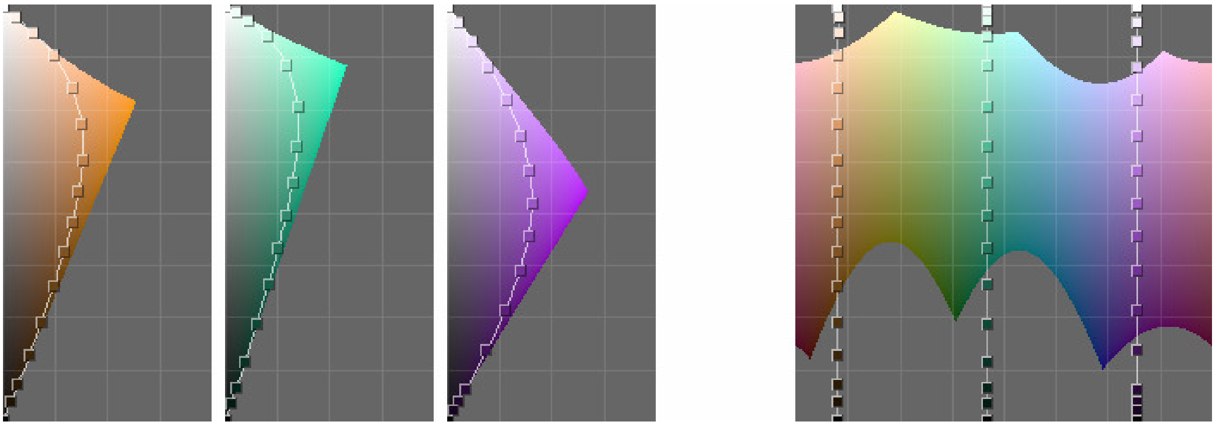


Fig. 5-16 Three hue slices and one saturation slice. They show the colors of three C_{seq} curves, using CIE94 as the distance function.

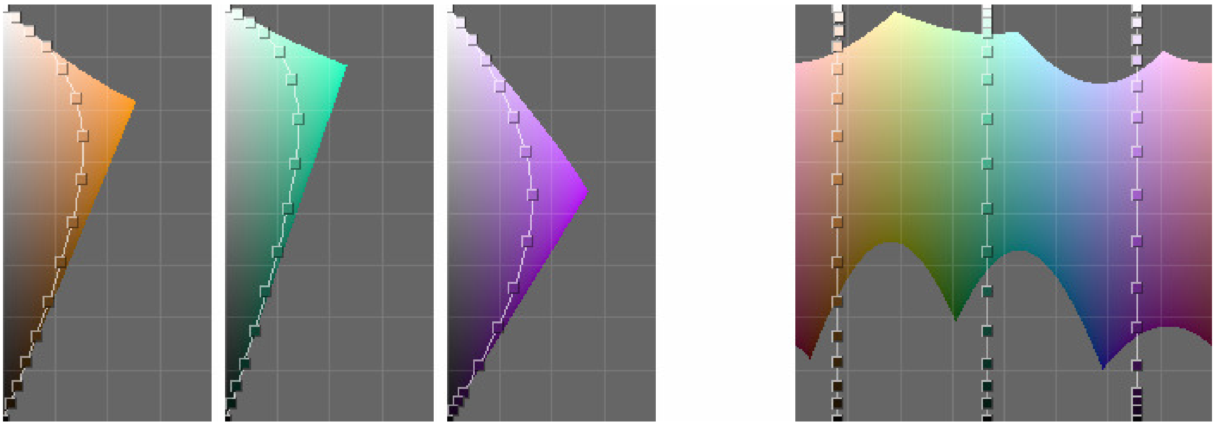


Fig. 5-17 Three hue slices and one saturation slice. They show the colors of three C_{seq} curves, using CIEDE2000 as the distance function.

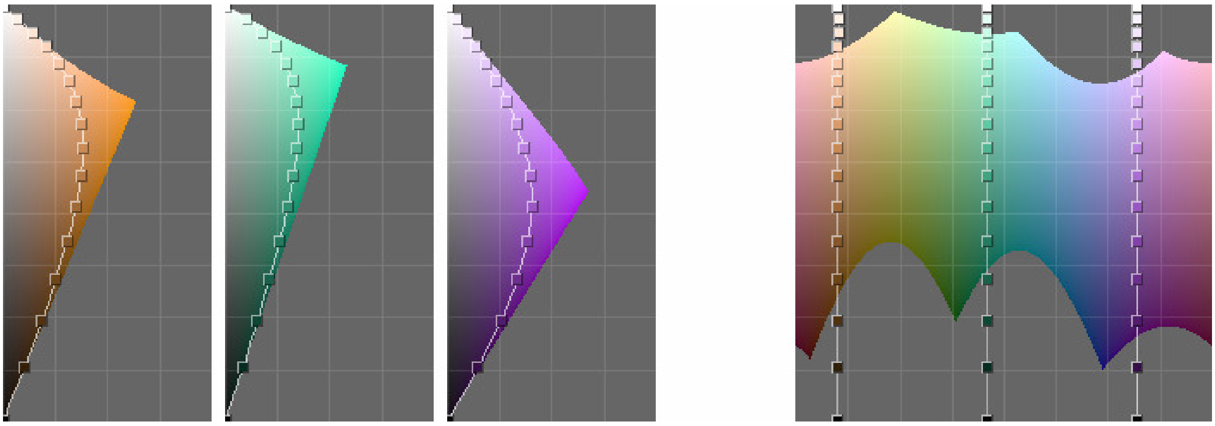


Fig. 5-18 Three hue slices and one saturation slice. They show the colors of three C_{seq} curves, using D as the distance function.

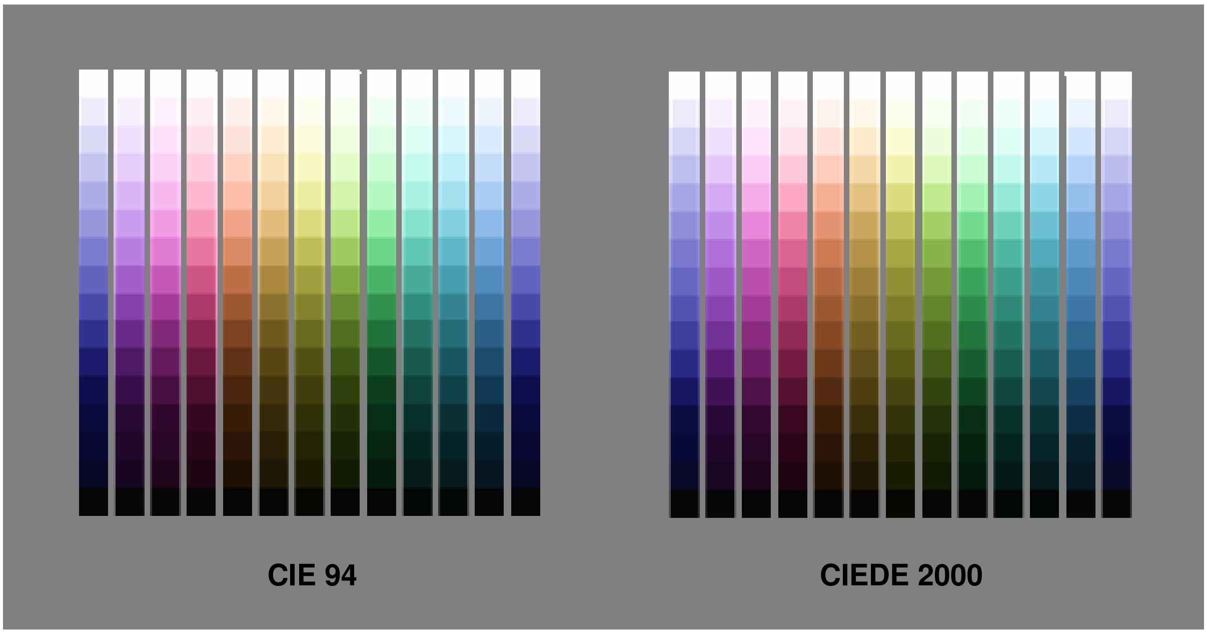


Fig. 5-19 Using the CIE94 or CIEDE2000 as distance functions yields palettes with colors that are not at equal perceptual distance to each other; the dark colors are too close to each other.

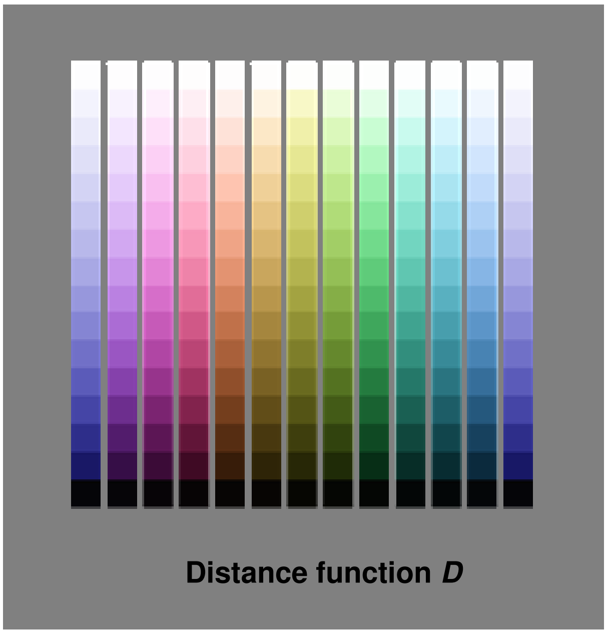


Fig. 5-20 The palettes created using *D* yield colors with more uniform lightness ordering.

5.6 Range on the curve

In section 5.3, the C_{seq} curve is defined that yields a continuous palette from black to white. The hue of the curve can be changed with parameter *h*, and the shape of the curve can be changed with *s*, which controls the saturation.

In this section, the colors on the curve are restricted to a certain range. By doing this the lightness contrast and the brightness of a palette can be adjusted.

We define t_0 to be the starting point of a palette and t_1 to be the end point. Parameter $c \in [0..1]$ which controls the lightness contrast can then be defined as

$$c = t_1 - t_0 .$$

The ranges of parameter t_0 and t_1 can now be defined in terms of *c*:

$$t_0 \in [0..1-c]$$
$$t_1 \in [c..1]$$

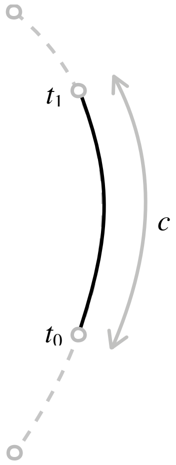


Fig. 5-21 Parameter *c* controls length of the curve. This determines the lightness contrast of the palette.

To control the position of range $[t_0..t_1]$ on the curve we introduce $b \in [0..1]$, which represents the brightness of the palette. We use *b* = 0 to denote the darkest palette and *b* = 1 to denote the lightest palette. We define t_0 in terms of *b* and *c* as

$$t_0 = (1-c)b.$$

Now input parameter $t \in [0..1]$ of palette P_{seq} can be transformed to range $[t_0..t_1]$ on curve C_{seq} by

$$P_{seq}(\sigma, t) = C_{seq}(\sigma,\ t_0 + t(t_1 - t_0)) .$$

which in terms of *b* and *c* is

$$P_{seq}(\sigma, t) = C_{seq}(\sigma,\ (1-c)b + tc) .$$

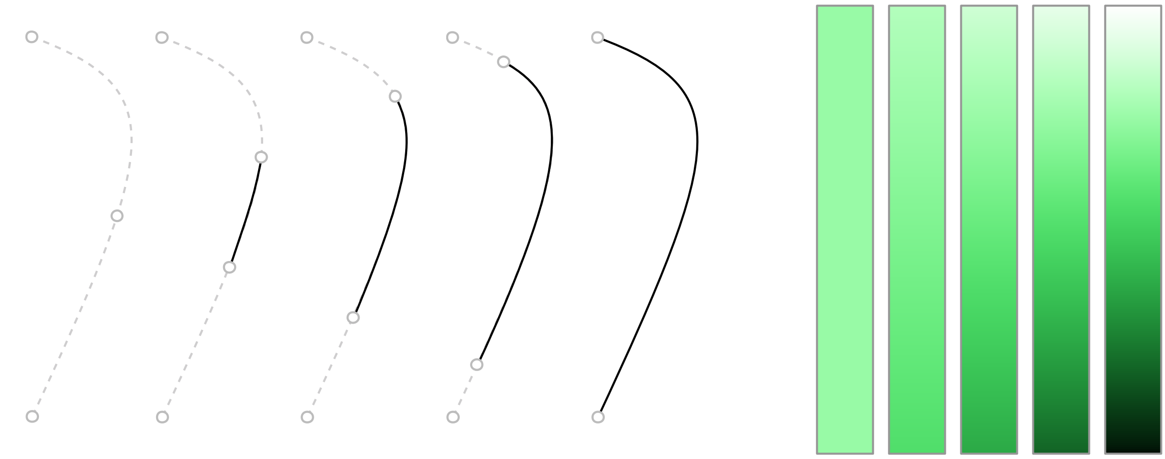


Fig. 5-23 Left: five P_{seq} curves with varying contrast: $c = \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$.
Right: the five corresponding palettes

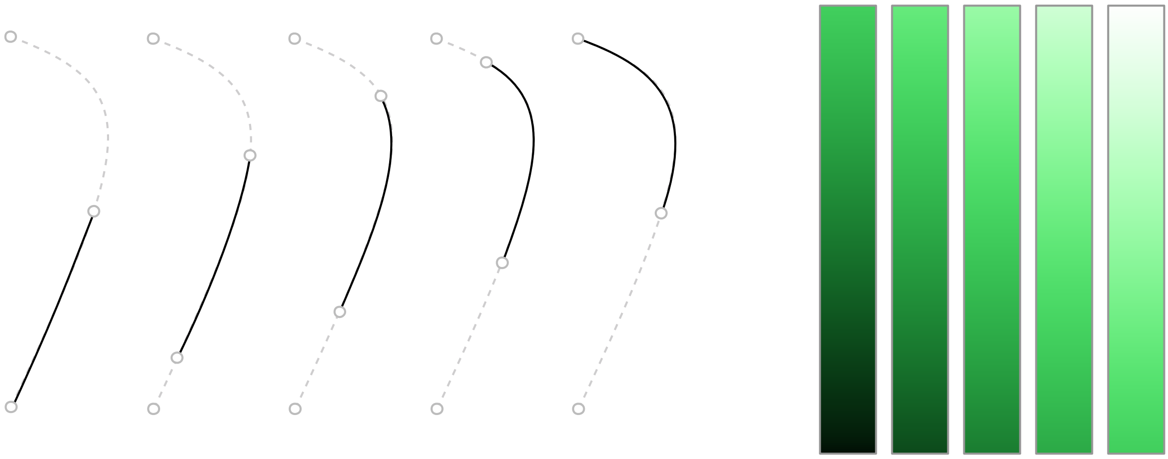


Fig. 5-23 Left: five P_{seq} curves with varying brightness: $b = \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$.
Right: the five corresponding palettes

5.7 Discrete palettes

In this section, the definition of continuous palette P_{seq} is extended to discrete palettes. We use the notation $P_{seq}(\sigma)[i]$ to denote the i^{th} color of palette $P_{seq}(\sigma)$, where $i \in [0,1,..,N-1]$.

For sequential palettes, we require all adjacent colors to be at equal perceptual distance:

$$\left(\forall_i : 0 \le i < N-2 : D\left(P_{seq}(\sigma)[i], P_{seq}(\sigma)[i+1]\right) = D\left(P_{seq}(\sigma)[i+1], P_{seq}(\sigma)[i+2]\right)\right).$$

T can be used to calculate points that are at equal arc distance from each other, since *T* returns the normalized arc length for a given *t*. However, the distance between two points is usually different from the arc difference between those two points (Fig. 5-24).

One way to find *N* points on a curve that are at equal distance is to use a force-based algorithm. All adjacent points exert a force on one another. The algorithm is comprised of two steps:

- Calculating the forces exerted on each point by its neighbor(s)
- Repositioning the points on the curve according to the total force applied on that point

This process is iterated until an acceptable finishing condition is reached.

However, because *D* only makes use of the lightness component, a simpler approach to get *N* points on the curve at equal distance can be used. We define $P_{seq}(\sigma)[i]$ by

$$P_{seq}(\sigma)[i] = P_{seq}\left(\sigma, \frac{i}{N-1}\right), \qquad 0 \le i < N .$$

Since the distance function only uses the *L*-component of a color, only one dimension needs to be taken into account. This reduces the problem of finding equidistant points that lie on a curve, to finding equidistant points along a straight line (Fig. 5-24). Therefore, this definition of $P_{seq}(\sigma)[i]$ yields points on the curve that are equally spaced according to *D*.

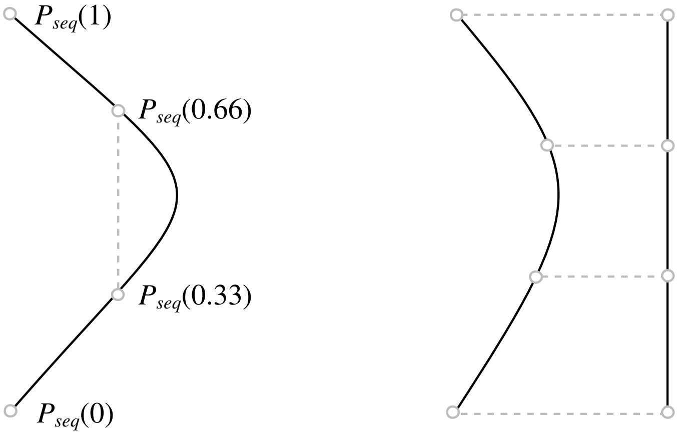


Fig. 5-24 Left: Normally, the arc distance between $P_{seq}(0.33)$ and $P_{seq}(0.66)$ is different from the normal distance. Right: since *D* only uses the L component, the arc distance is equal to the normal distance.

The Brewer palettes display a relation between the overall lightness contrast and the number of colors of a discrete palette. We evaluated the Brewer palettes in CIELUV color space and compared them with palettes defined by P_{seq}. The palettes of Brewer display an approximately linear increase in lightness contrast as N is increased. The following relation is used

$$b = 0.25 \text{ and } c = \min(0.88, 0.34 + 0.06N) .$$

`

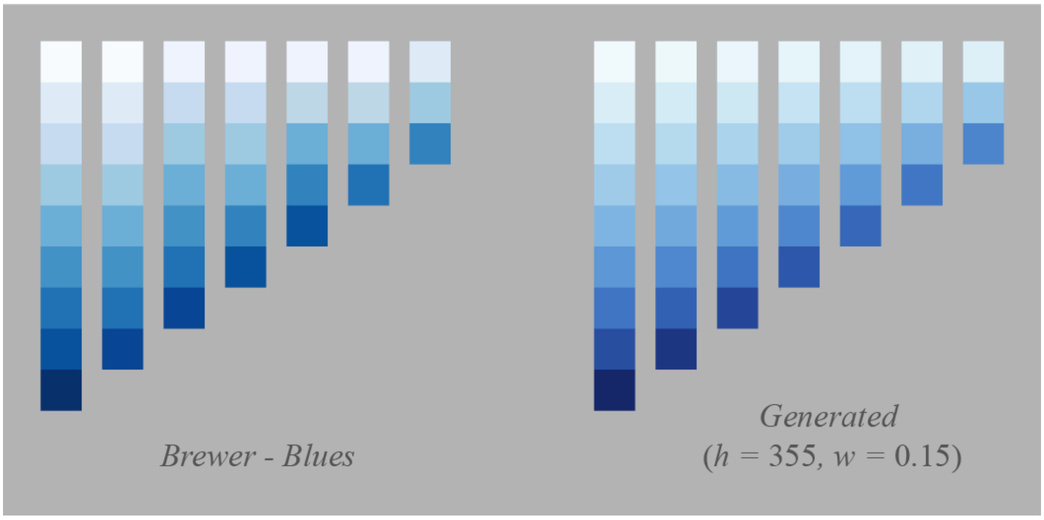


Fig. 5-25 Seven Brewer palettes and seven generated palettes, they display similar increases in lightness contrast as N is increased.

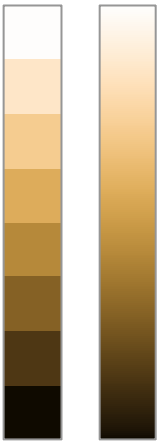


Fig. 5-26 A discrete palette with its corresponding continuous palette.

5.8 Multi-hue sequential palettes

In this section, we extend the P_{seq} definition to a multi-hue palette model.

The multi-hue palettes defined by Cynthia Brewer are shown below. They can be categorized in three groups, shown in Fig. 5-27. When the colors of a palette get lighter, the palette changes gradually towards a certain bright point. The three groups each have a bright point in common. This does not mean that all palettes end in their corresponding bright point, but the hues progress in their direction. This bright point convergence is illustrated by Fig. 5-27. Note that the three views are all shifted in hue to put the bright point hue in the middle of the picture.

Although the model we propose later in this section can be used to generate multi-hue palettes of all three groups, we have a preference for the yellow-topped palettes. There are several reasons for this.

The first reason is purely esthetical; we find that the color combinations of the cyan- and pink-topped palettes do not yield as nice looking palettes as the yellow-topped palettes.

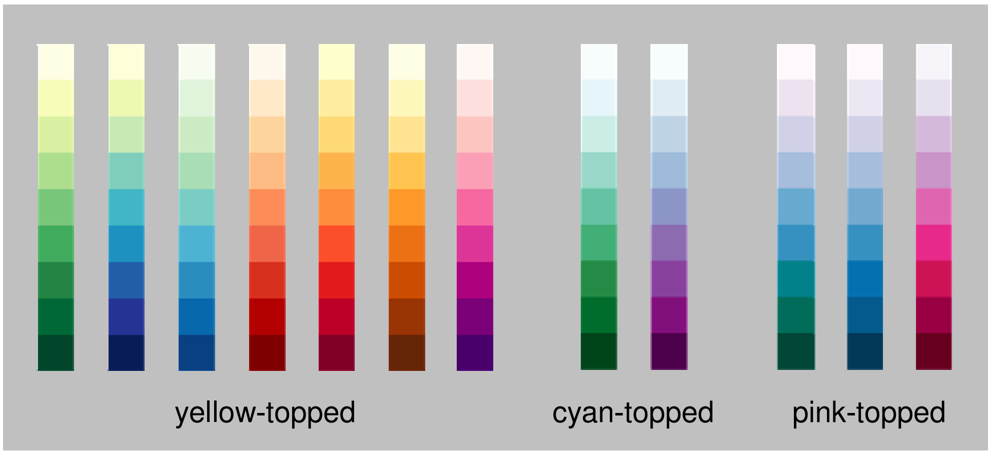


Fig. 5-27 The multi-hue sequential palettes divided in three categories.

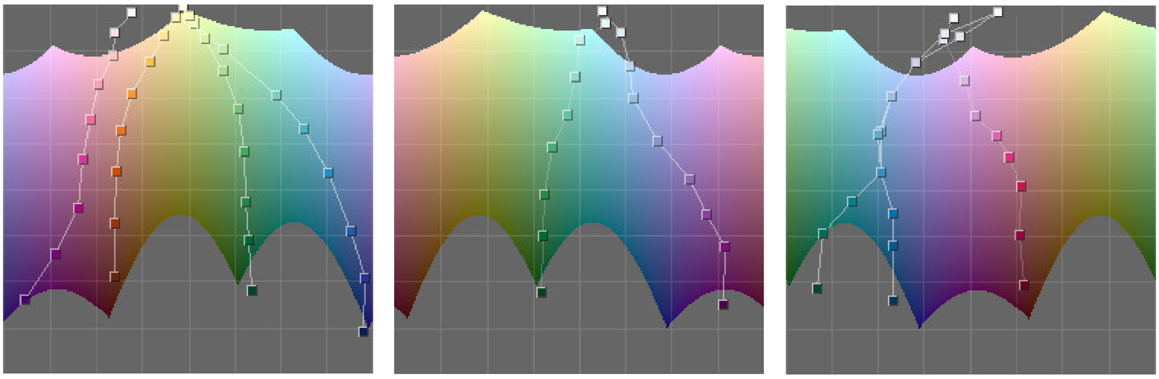


Fig. 5-28 Three saturation slices showing the yellow-topped, cyan-topped and pink-topped palettes. The hues converge to a certain bright point hue.

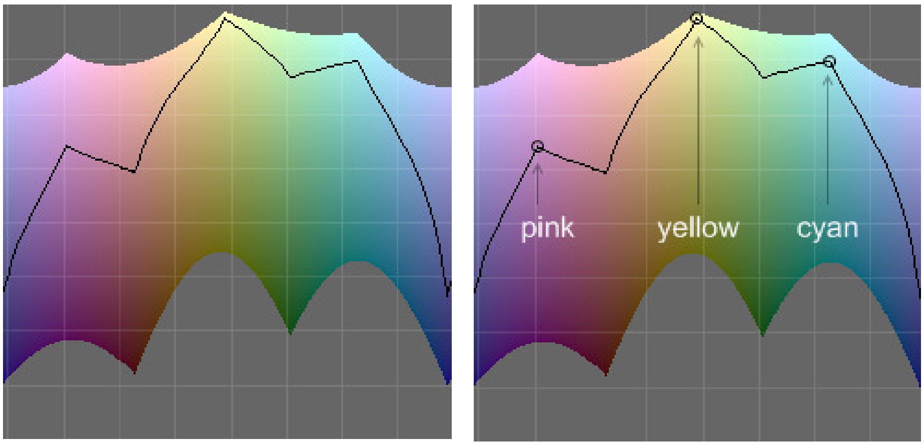


Fig. 5-29 A saturation slice with a projection of the path of all MSCs. The yellow MSC has the highest lightness.

Furthermore, the yellow hue allows lighter colors to be more saturated. This is illustrated by Fig. 5-29, which shows the lightness values of the MSCs of all hues. The three bright points to which the Brewer palettes converge are local maxima of the MSC path. However, the yellow MSC has the highest lightness; this allows the lighter colors to be more saturated for the hues around yellow than for other hues.

Finally, most multi-hue sequential palettes defined by Brewer are yellow-topped. Even the hues of the single-hue sequential palettes slightly move towards yellow for the lighter colors. Furthermore, Brewer only defined multi-hue diverging palettes with a yellow bright point (Fig. 5-30). So her palettes also display a preference for the yellow bright point.

For the reasons explained above, we choose yellow as the default bright point for the multi-hue palettes. A straightforward approach for creating multi-hue palettes would be to change the hue of the top point of a palette to the bright point hue. Then the sequential palette requirements are satisfied, but it does not exploit the saturation potential of the bright point hue. The colors in the lighter part of the palette look relatively dull and unsaturated, while they could have had a higher saturation.

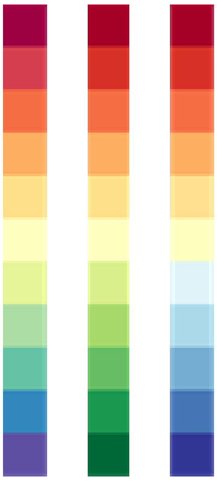


Fig. 5-30 The multi-hue diverging palettes defined by Brewer only use a yellow bright point.

Another straight-forward method is to use colors of several sequential palettes with varying hue to construct a multi-hue palette. This, however, creates discontinuities in the first order derivative of the curve, and yields irregular saturation transitions.

Therefore, our solution is to move the top point p_2 towards the bright point. Multi-hue variable w controls how close the top point is to the bright point. By moving the top point towards the bright point the saturation for the lighter colors is increased, yielding a more vibrant and colorful palette. Furthermore, the curve stays smooth this way, leading to a regular transition in saturation. The complete definition of the new top point is given below.

Multi-hue sequential palette definition

$$
\begin{aligned}
p_2.L &= (1-w)\cdot 100 + w\cdot p_b.L \\
p_2.S &= \min(\mathrm{S}_{\max}(p_2.L,\ p_2.H),\ \ w\cdot s\cdot p_b.S) \\
p_2.H &= \mathrm{MixHue}(w,\ \ h,\ \ p_b.H)
\end{aligned}
$$

where MixHue and $\mathrm{S}_{\max}$ are defined below.

Default values:
$w = 0.15$
$p_b = LCH(RGB(1,1,0))$ *// yellow*

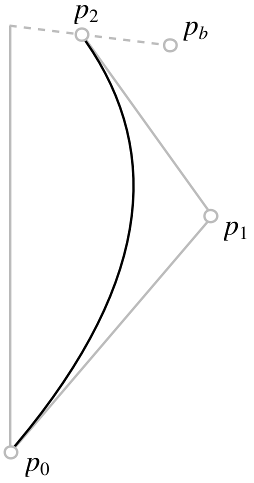


Fig. 5-31 p_2 is moved towards the bright point p_b as w is increased.

The p_2 point is moved towards the bright point proportionally to w. The MixHue function mixes the hues using the shortest path on the color circle. For example, MixHue(0.5, 340, 50) yields 15.

Definition of MixHue

$$\mathrm{MixHue}(\alpha, h_0, h_1) = \left(h_0 + \alpha M\right) \bmod 360\,,$$

where

$$M = (180 + h_1 - h_0) \bmod 360 - 180\,.$$

The resulting color p_2 can lie outside displayable color space. Therefore, $\mathrm{S}_{\max}$ calculates the maximum saturation for a color of a certain lightness and hue. Using that saturation value, the color is clamped back to the triangular surface, and its lightness and hue value are preserved.

Definition of $\mathrm{S}_{\max}$

$$\mathrm{S}_{\max}(l,h') = \alpha(p_{mid}.S - p_{end}.S) + p_{end}.S,$$

where

$$\alpha = \frac{p_{end}.L - l}{p_{end}.L - p_{mid}.L}$$

$$p_{mid} = \mathrm{MSC}(h')$$

$$p_{end} = \begin{cases} \mathrm{LCH}(0,0,0), & l \le p_{mid} \\ \mathrm{LCH}(100,0,0), & l > p_{mid} \end{cases}$$

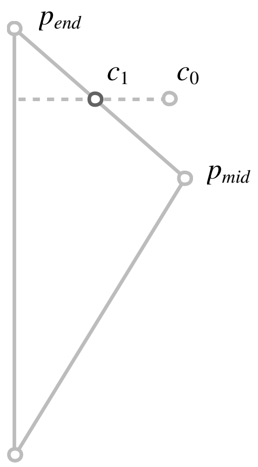


Fig. 5-32 Color c_0 is clamped back to displayable color space, using $\mathrm{S}_{\max}$.

Moving colors towards yellow makes their appearance warmer and moving them away colder. Therefore, we call the w parameter cold/warm.

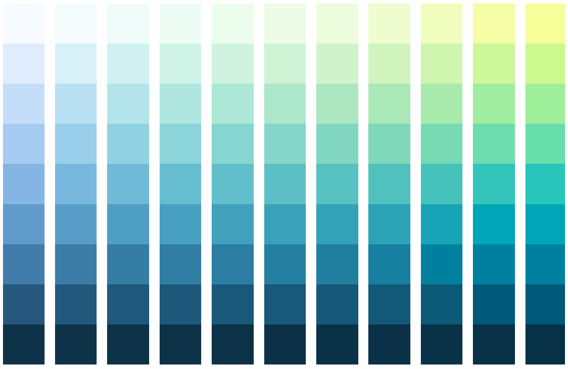


Fig. 5-33 Eleven palettes with w values from 0 to 1. As w is increased the palette's appearance looks warmer.

6 Diverging palettes

In the previous chapter, we presented a model for generating sequential palettes. In this chapter, the sequential palettes are used to generate diverging palettes, which can be altered using the same parameters as the sequential palette.

For completeness and reference, the complete diverging palette model is given on the next page.

6.1 Diverging palette properties

The diverging palette properties differ from the sequential palette properties, in that they have two hues, h_0 and h_1, and there is an additional parameter $m_{¸}$ which represents the midpoint position.

The h parameter is split in h_0 and h_1, because the hue is the only property that we define separately for each side of the diverging palette. The other properties are used for both sides of the palette. Keeping the b and c parameter equivalent for both sides and setting w to 0 takes care that colors at equal distance from the midpoint have the same lightness value:

$$\left(\forall x, \sigma_{div} : x \in [0..\min(m, 1-m)] \wedge \sigma_{div} \in \Sigma_{div} : P_{div}(\sigma_{div}, m-x).L = P_{div}(\sigma_{div}, m+x).L\right).$$

The range of x is defined as $[0..\min(m, 1\text{-}m)]$ because the midpoint is variable. The above equation is satisfied, because the distance function D is solely based on lightness. Therefore, the same t value yields colors of equal lightness for sequential palettes with equal b, c and w = 0. Note that for multi-hue palettes the top point is shifted towards a color with lower L-value, resulting in slightly lower L-values for all colors in the palette.

6.2 Two sequential palettes

The diverging palette is comprised of two sequential palettes and a possible neutral color c_n in the middle:

$$P_{div}(\sigma_{div}, t) = \begin{cases} P_{seq}(\sigma_0, 2t'), & t' < 0.5 \\ P_{seq}(\sigma_1, 2(1-t')), & t' > 0.5. \\ c_n, & t' = 0.5 \end{cases}$$

A consequence of using the sequential palettes like this is that the resulting palette is not continuous for $b \neq 1$. However, the discontinuity occurs at the midpoint. Since the two parts of the diverging palette should be distinguishable from each other a discontinuity should be expected.

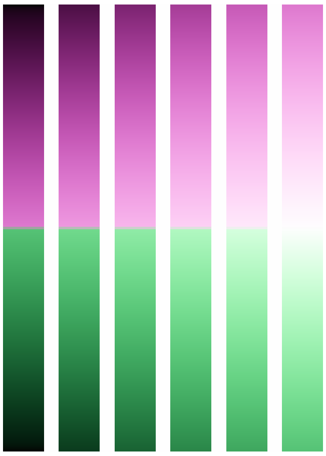


Fig. 6-1 The discontinuity at the midpoint shows clearly for the palettes with $b \neq 1$.

Diverging palette definition

Palette properties:

$\Sigma_{div} = [1..\infty] \times [0..360] \times [0 \times 360] \times [0..1] \times [0..1] \times [0..1] \times [0..1] \times [0..1]$

$\sigma_{div} : \Sigma_{div}$

$\sigma_{div} = \langle N, h_0, h_1, w, m, c, s, b \rangle$

Palette:

$P_{div} : \Sigma_{div} \times [0..1] \mapsto LCH$

$$P_{div}(\sigma_{div}, t) = \begin{cases} P_{seq}(\sigma_0, 2t'), & t' < 0.5 \\ P_{seq}(\sigma_1, 2(1-t')), & t' > 0.5, \\ c_n, & t' = 0.5 \end{cases}$$

where

$\sigma_0 = \langle N, h_0, w, c, s, b \rangle$

$\sigma_1 = \langle N, h_1, w, c, s, b \rangle$

$$t' = \begin{cases} \dfrac{t}{1+|2\varepsilon|} & m \geq 0.5 \\ \dfrac{t+|2\varepsilon|}{1+|2\varepsilon|} & m < 0.5 \end{cases}$$

$\varepsilon = m - 0.5.$

Neutral color definition:

$c_n.L = \dfrac{c_0.L + c_1.L}{2}$

$c_{n.}S = \min(\ S_{\max}(c_n.L, c_n.H),\ s_n\ )$

$c_n.H = p_b.H$

$s_n = \dfrac{c_0.S + c_1.S}{2} w$

$c_0 = P_{seq}(\sigma_0, 1)$

$c_1 = P_{seq}(\sigma_1, 1)$

Default values:

$w = 0$

$m = 0.5$

$c = \min(0.88,\ 1.0 - 0.06(11 - N'))$, where $N' = (N \text{ div } 2) + 1$

$s = 0.6$

$b = 0.75$

6.3 Midpoint position

In this section we elaborate on how the midpoint is moved within the diverging palette by parameter $m \in [0..1]$. m = 0 denotes that the midpoint lies at the beginning of the palette, m = 1 denotes that the midpoint lies at the end. The midpoint can lie anywhere in between, but m = 0.5, i.e., the midpoint is in the middle, is the default value. Note that if m = 0 or m = 1 the diverging palette looks like a sequential palette.

The midpoint is moved in such a way that the equation defined in section 6.1 holds, as shown in Fig. 6-3. Moving the midpoint this way is essentially the same as using only a certain range of the original diverging palette that has a midpoint in the middle. The new parameter t' should therefore yield $t' = 0.5$ for $t = m$. Variable ε is the directed distance between m and the original midpoint 0.5:

$$\varepsilon = m - 0.5 \ .$$

To put the new midpoint m in the center, the range of t is increased by $|2\varepsilon|$. The result is scaled back to the range [0..1] to get the corresponding t'.

We distinguish between two cases, shown in Fig. 6-3.

- If $m \geq 0.5$, the midpoint is shifted up. Then $0 \ \leq \ t' \ \leq \ \dfrac{1}{1+|2\varepsilon|}$.
- If $m < 0.5$, the midpoint is shifted down. Then $\dfrac{|2\varepsilon|}{1+|2\varepsilon|} \ \leq \ t' \ \leq \ 1$.

The definition below defines the transformation of t to get the desired t' of the palette with the midpoint shifted to m:

$$\begin{aligned} t' &= \begin{cases} \dfrac{t}{1+|2\varepsilon|} & m \geq 0.5 \\ \dfrac{t+|2\varepsilon|}{1+|2\varepsilon|} & m < 0.5 \end{cases} \\ \varepsilon &= m - 0.5. \end{aligned}$$

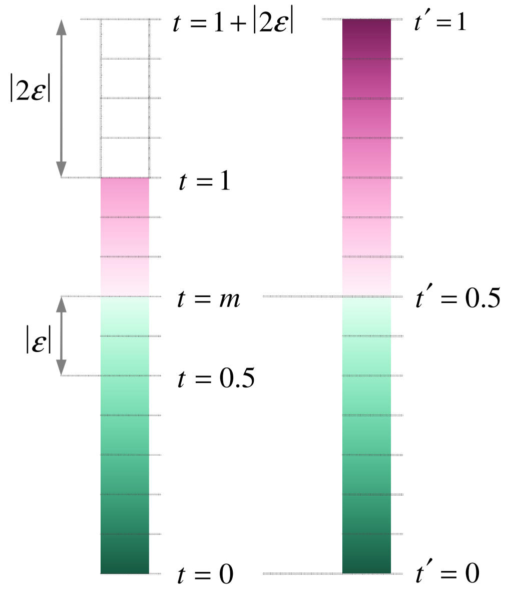


Fig. 6-3a. $m > 0.5$. The right palette is the original, with $m = 0.5$. t is scaled such that the left palette has $m = 0.7$, $\varepsilon = 0.2$.

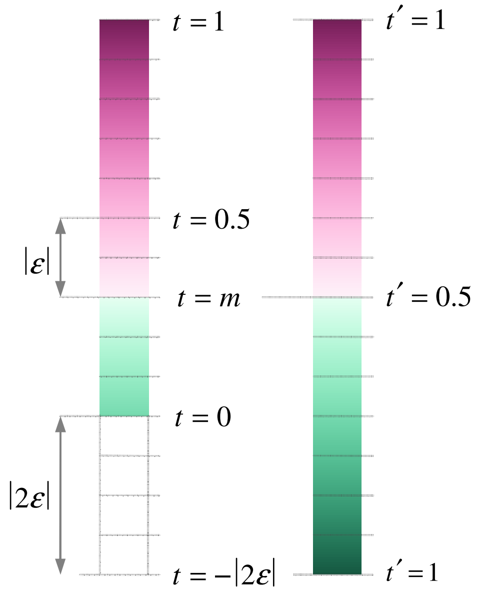


Fig. 6-3b. $m < 0.5$. The right palette is the original, with $m = 0.5$. t is scaled such that the left palette has $m = 0.3$, $\varepsilon = -0.2$.

6.4 Discrete diverging palettes

When generating a discrete diverging palette, the neutral color needs to be taken in account. In Fig. 6-4, two Brewer palettes are shown. The left palette has the exact same colors as the right, except for the neutral color.

The generated palettes should show the same: if a palette with N colors has no neutral color, then it should use the colors from a neutral color palette with $N+1$ colors except for the neutral color. If this would not be done, the difference between the two colors around the midpoint would be relatively small as can be seen in the left palette of Fig. 6-5. For the right palette this difference is larger, and the difference between the other colors is smaller, which yields a more balanced palette.

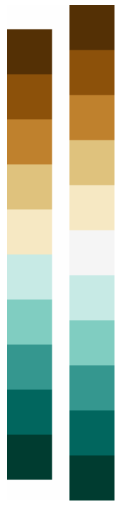


Fig. 6-4 Two diverging Brewer palettes. The left has the exact same colors as the right, except for the neutral color.

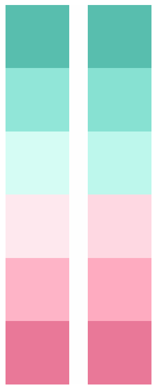


Fig. 6-5. The left palette is generated by $P_{div}\left(\sigma'_{div}, \frac{i}{N-1}\right)$. The right one is generated by the definition described below.

Discrete palette:

$$P_{div} : \Sigma_{div} \times [0,1..,N) \mapsto LCH$$

$$P_{div}(\sigma_{div})[i] = \begin{cases} P_{div}\left(\sigma'_{div},\ \dfrac{i}{N-1}\right), & 2m' \bmod 2 \ = \ 0 \quad \text{// With neutral color} \\ P_{div}\left(\sigma''_{div},\ \dfrac{i}{N}\right), & 2m' \bmod 2 \ \neq \ 0 \wedge i < m' + 0.5 \quad \text{// Without neutral color} \\ P_{div}\left(\sigma''_{div},\ \dfrac{i+1}{N}\right), & 2m' \bmod 2 \ \neq 0 \wedge i \geq m' + 0.5 \quad \text{// Without neutral color} \end{cases}$$

$$m' = \frac{\lfloor 2m(N-1) + 0.5 \rfloor}{2}$$

$$\sigma'_{div} = \left\langle N, h_0, h_1, w, \frac{m'}{N-1}, c, s, b \right\rangle$$

$$\sigma''_{div} = \left\langle N+1, h_0, h_1, w, \frac{(m'+0.5)}{N}, c, s, b \right\rangle$$

For a Brewer palette it is simple to detect whether it has a neutral color: if *N* is odd, then the palette has a neutral color, otherwise it does not. Unlike the Brewer palettes, our palettes have a variable midpoint. To be able to check whether our palettes have a neutral color, the normalized midpoint position *m* is transformed to a discrete midpoint position by

$$m' = \frac{\lfloor 2m(N-1) + 0.5 \rfloor}{2}.$$

This takes care that the discrete midpoint position is in the range [0, 0.5, … , N]. Fig. 6-6 shows where the midpoints could lie for a palette with $N = 5$. If m' is an integer number, then the palette has a neutral color, otherwise it does not.

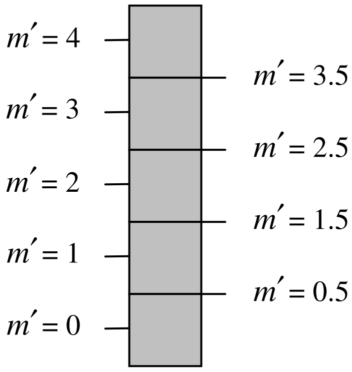


Fig. 6-6 The possible discrete midpoint positions for a 5 color palette.

If a neutral color is present, i.e., if $2m' \bmod 2 = 0$, then the translation from continuous to discrete is straightforward:

$$P_{div}(\sigma_{div})[i] = P_{div}\left(\sigma'_{div}, \frac{i}{N-1}\right),$$

where

$$\sigma'_{div} = \left\langle N, h_0, h_1, w, \frac{m'}{N-1}, c, s, b \right\rangle.$$

Note that the continuous diverging palette is passed the normalized m'. If m would be used, then it could be the case that its corresponding discrete midpoint is not in the range [0, 0.5, … , N]. This would result in unbalanced diverging palettes, in which one side of the palette is darker then the other side.

If the palette does not have a neutral color, i.e., if $2m' \bmod 2 \neq 0$, the definition is as follows

$$P_{div}(\sigma_{div})[i] = \begin{cases} P_{div}\left(\sigma''_{div}, \dfrac{i}{N}\right), & i < m' + 0.5 \\ P_{div}\left(\sigma''_{div}, \dfrac{i+1}{N}\right), & i \geq m' + 0.5 \end{cases}$$

where

$$\sigma''_{div} = \left\langle N+1, h_0, h_1, w, \frac{(m'+0.5)}{N}, c, s, b \right\rangle.$$

This definition treats the palette as if N is $N+1$ and the discrete midpoint $m' := m' + 0.5$. If the desired color number i is smaller than the discrete midpoint, i is used; otherwise i is increased by one, to skip the midpoint.

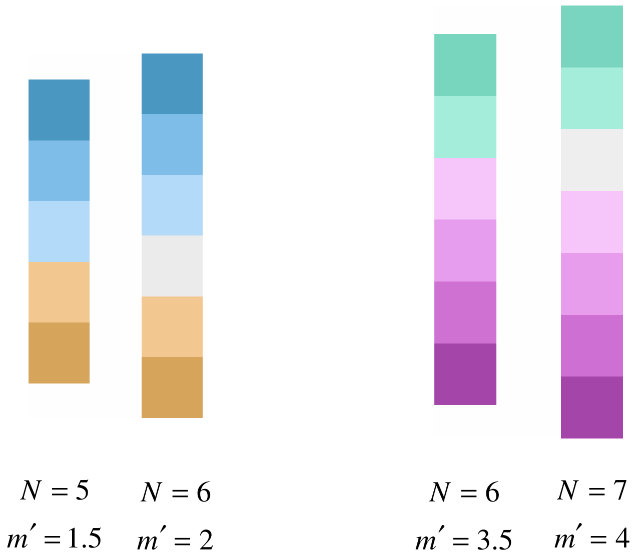


Fig. 6-7. Two groups of generated palettes. The left palette has no neutral color, the right palette does. Odd and even palettes can have a neutral color in our representation.

Finally, the default contrast value for diverging palettes is defined to be equal to the sequential palettes, except that $N' = (N \text{ div } 2) + 1$. The presence of a neutral color is not used in the equation, since we do not want the default contrast to change when m is varied.

$$c = \min(0.88,\ 1.0 - 0.06(11 - N')), \quad \text{where } N' = (N \text{ div } 2) + 1$$

6.5 The neutral color

The neutral color is usually a light grayish color. However, for multi-hue palettes this does not need to be the case, as the Brewer palettes in Fig. 6-8 show. The neutral color can also be the brightpoint color. Therefore, we define neutral color c_n as folows

$$\begin{aligned} c_n.L &= \frac{c_0.L + c_1.L}{2}, \\ c_n.S &= \min\left(\ \mathrm{S}_{\max}(c_n.L, c_n.H),\ s_n\ \right), \\ c_n.H &= p_b.H, \end{aligned}$$

where

$$\begin{aligned} s_n &= \frac{c_0.S + c_1.S}{2} w, \\ c_0 &= P_{seq}(\sigma_0, 1), \\ c_1 &= P_{seq}(\sigma_1, 1). \end{aligned}$$

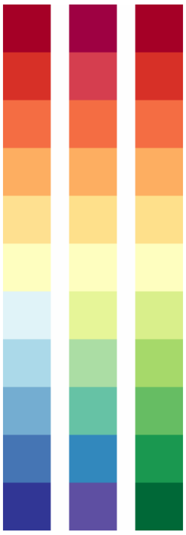


Fig. 6-8. The three multi-hue diverging palettes of Cynthia Brewer.

The lightness of the midpoint is determined by the average of the lightness of the top points of the two sequential palettes. An average is used because sequential palettes of similar w can still have small differences in lightness of the top point.

The hue of the midpoint is always set at the bright point hue.

The saturation then determines whether the midpoint is grayish or yellowish, i.e., whether saturation is low or high. s_n is the average of the saturation values of c_0 and c_1. This is multiplied with w, because c_0 or c_1 are not always desaturated if w = 0. Since the c_n could yield a color outside displayable color space, the minimum with $\mathrm{S}_{\max}$ is used to clamp it back.

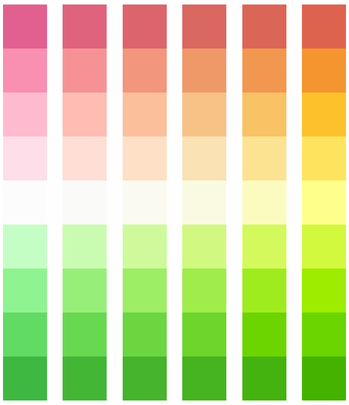


Fig. 6-9. Six multi-hue diverging palettes of which w = {0, 0.2, 0.4, 0.6, 0.8, 1}.

6.6 Extension to bivariate palettes

Just as the sequential palettes are used as a basis for diverging palettes, they can be used to generate bivariate color palettes. To form the bivariate palette, M sequential palettes of different hue are put together.

The colors of sequential palettes have colors of equal lightness, for palettes of equal b and c, and $w = 0$. Therefore, our sequential palettes generated by P_{seq} are well suited to be used for generating bivariate palettes.

For proper bivariate palettes, a relatively high contrast and low M and N can be recommended, since bivariate palettes are usually hard to interpret when used in a visualization.

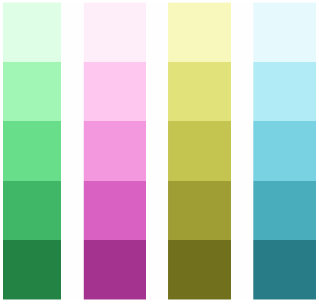


Fig. 6-10. An example of four sequential palettes put together to form a bivariate palette with $M = 4$ and $N = 5$.

7 Qualitative palettes

In this section, we present two different types of qualitative palette models which are each better suited for different situations.

The requirements for qualitative palettes are:

- All colors are distinguishable.
- The palette displays no clear ordering.
- No color stands out between the orders; all colors are perceived as equally important.
- Preferably colors are nameable.

Fig. 7-1 shows the qualitative palettes that Brewer defined. It is clear that these palettes do not follow such strict rules as the sequential and diverging palettes, which all look alike more. Each of these qualitative palettes has its own strong points.

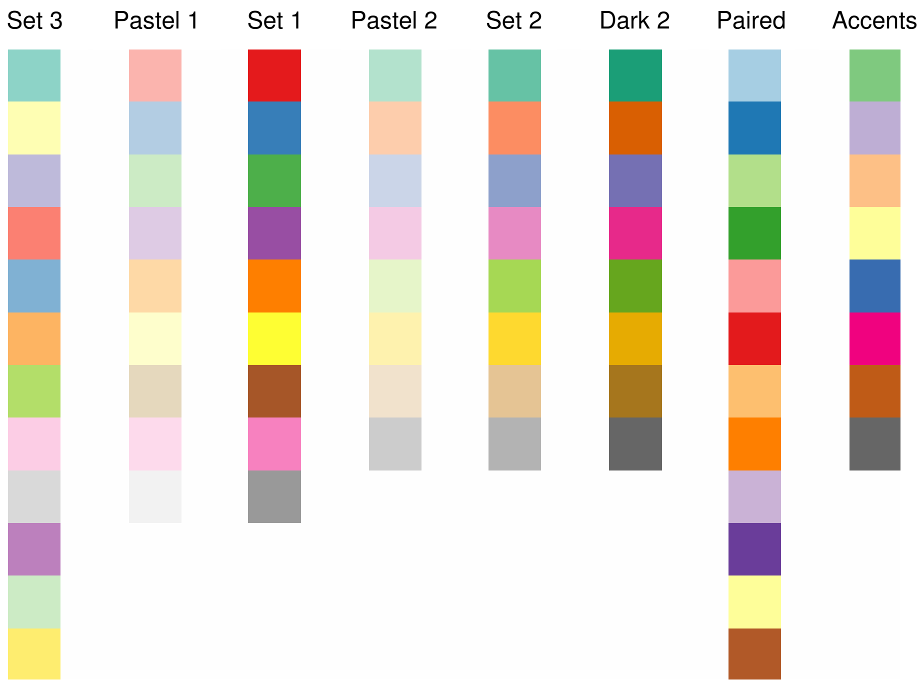


Fig. 7-1 The eight qualitative palettes defined by Brewer

The only palettes that satisfy the nameability requirement are Set 1 and Pastel 1, although for the latter the nameability is lower, due to the low saturation and high lightness. Set 3 and Paired are the only ones with more than nine colors; however, they do not satisfy the equal-importance requirement. For Set 3, some colors are clearly more saturated than others, for Paired there is a clear brightness difference.

In the table below, we rated the palettes according to the given requirements; + means positive, o neutral, and - negative.

The distinguishability of Set 3 and Paired are rated lower than the rest, but this is mostly due to the fact that more colors are used. The distinguishability of Dark 2 is also rated as 'o'; the colors of the Brewer palettes are arranged such that the distinguishability appears high, however, when the colors are rearranged, more ordering shows and less distinguishability. For Dark 2 distinguishability is lowest; for the other two it is higher, because not all colors increase equally in lightness.

Apart from the nameability requirement, the values in the table are subjective. However, the aim of this table is to show that is hard, maybe impossible, to generate a palette that satisfies all desired requirements.

Furthermore, a subjective requirement is that the palette is aesthetically satisfying. Set 1 seems to score very high in the table, but we found its collection of very saturated colors not really aesthetically pleasing.

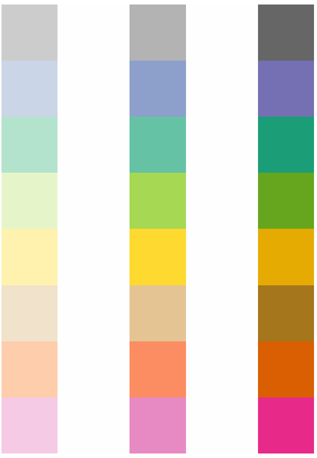


Fig. 7-2 Pastel 2, Set 2 and Dark 2 with rearranged colors show lower distinguishability and more ordering.

	Set 3	Pastel 1	Set 1	Pastel 2	Set 2	Dark 2	Paired	Accents
Distinguishability	o	+	+	o	+	o	o	+
No ordering	+	+	+	+	+	+	+	+
Equal-importance	-	o	o	o	+	o	-	-
Nameability	-	o	+	-	-	-	-	-

Another problem is that we want to be able to control the appearance of palettes, in order to make them suited for different visualizations. In other words, we want the brightness, contrast and saturation to be variable. To keep the nameability of colors invariant under these changes is hard, because our perception of colors is not only dependent on hue. Light red is perceived as pink, dark yellow or dark orange is perceived as brown. The hue between blue and cyan is usually called 'light blue' which overlaps with the name of the light colors of the blue hues.

Therefore, our conclusion is that it is impossible to satisfy all these requirements at once and to still end up with an aesthetically pleasing palette. Fortunately, most of the time, not all requirements are important to the user.

We therefore define two ways of generating palettes. The first palette has fixed colors, so nameability can be guaranteed. This palette can be optimized to have high distinguishability, no ordering and nameable colors. The second palette has variable colors that can be tuned using certain parameters, but nameability can not be guaranteed.

7.1 Fixed-color palette

The use of a fixed-color palette makes it possible to optimize a palette to fit certain requirements. Of course the colors of one of the Brewer palettes can be used [5][6]. However, we designed a qualitative palette that has 17 colors. This fixed-color palette is defined as follows in this ordering:

🟦	RGB(92, 107, 247),	blue	blauw
🟥	RGB(255, 89, 89),	red	rood
🟩	RGB(92, 203, 92),	green	groen
🟧	RGB(255, 177, 17),	orange	oranje
🟪	RGB(170, 97, 187),	purple	paars
🟨	RGB(255, 255, 95),	yellow	geel
🟪	RGB(255, 137, 235),	pink	roze
🟫	RGB(145, 101, 62),	brown	bruin
⬜	RGB(193, 193, 193),	gray	grijs
🟦	RGB(92, 229, 214),	aqua	turquoise, blauwgroen
🟩	RGB(201, 255, 135),	lime green	limoen groen
⬜	RGB(255, 224, 204),	skin pink	huidskleur
🟥	RGB(173, 45, 92),	Bordeaux red	Bordeaux rood
🟪	RGB(227, 196, 239),	lilac, light purple	lila paars
🟨	RGB(226, 212, 149),	beige	beige
🟦	RGB(204, 241, 255),	light blue	licht blauw
🟩	RGB(87, 142, 82),	dark green	donker groen

These colors are chosen to maximize distinguishability, display no ordering and to still be nameable. Firstly, note that we defined these colors in such a way as to yield nameable colors in Dutch, the colors skin pink and lilac are less accepted color terms in English than they are in Dutch. For that reason we added the Dutch names as well. Furthermore, this palette was purposely ordered. For an *N* color qualitative palette, we suggest to use the first *N* colors.

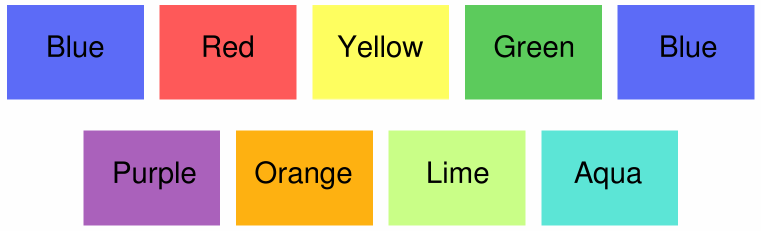


Fig. 7-3 The four base colors blue, red, yellow, and green are showed on the top row.
The results of mixing two colors are shown on the bottom row.

The first eight colors of the fixed-color palette are the generally accepted unique hues. They look very much like Set 1, with the difference that these colors are slightly lighter to make the difference with the darker brown color somewhat higher. These colors are followed by aqua and lime. If red, green, blue and yellow are the most important colors, then aqua and lime are the logical colors that follow because they fall in between (Fig. 7-3). So although they have not as clearly defined names, aqua and lime are the 10th and 11th color. Skin pink, Bordeaux red, lilac and beige were chosen because of their relatively well-known names in Dutch, although they are not as clear as the unique hues. And finally, light blue and dark green were added to have two more distinguishable colors.

Fixed-color qualitative palette evaluation:

- **Distinguishability:** This palette yields relatively distinguishable colors, especially the first ten colors. The more colors are added, the lower distinguishability gets.

- **No ordering:** There is no clear ordering on this palette.

- **Equal-importance:** For the first seven colors we find that no color seems more important than the others. However, to maintain high enough distinguishability, less saturated colors are also used, which appear less important than the more saturated ones. Therefore, this palette does not satisfy the equal-importance requirement for $N > 7$.

- **Nameability:** The nameability is guaranteed for the first eight colors. After those, we tried to put together nameable colors, which use distinctive names. However, we acknowledge that these names can never be as clear as the first eight colors.

This fixed-color palette yields a large set of colors, which can be useful for many visualization applications, since sometimes 12 is just not enough. And even for 17 colors, they have a fairly high distinguishability. However, it must be noted that the fewer colors used, the better the palette fits all requirements.

7.2 Generated palette

In this section we present our procedurally generated qualitative palette.

Our aim is to generate a palette that is aesthetically pleasing, while still being able to alter its appearance. For sequential and diverging palettes, this is relatively easy; if only two base hues are combined, then it is hard to create an ugly color combination. A qualitative palette is typically a combination of colors of differing hue. To get a set of colors that are in harmony with each other is much harder and much more susceptible to taste.

A way to generate a qualitative palette with no ordering and with colors of equal importance is to use the approach suggested by Kurt Hornik and Achim Zeileis [32], shown in Fig. 7-4. They take *N* colors of equal saturation and lightness and vary the hue. This yields a palette without ordering and no color appears more important than the others. This method does not allow for a great number of distinguishable colors. They use a rather low saturation so no colors need to be clamped. Therefore, the color distance between the colors in the palette is relatively low. Furthermore, we find that their qualitative palettes do not look very aesthetically pleasing.

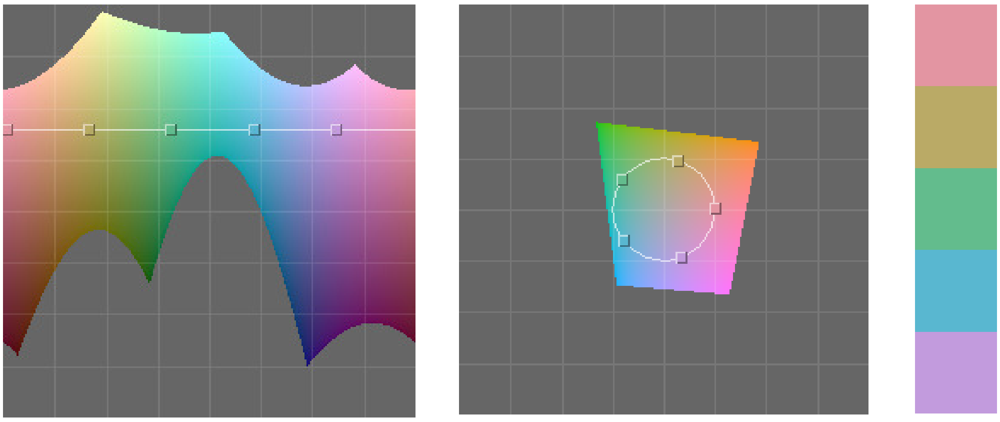


Fig. 7-4 A qualitative palette generated via the method proposed by Achim Zeileis and Kurt Hornik. The colors are of equal lightness and saturation, the hue is varied. The saturation is kept low to avoid color clamping.

We analyzed these palettes to find out why they were not aesthetically pleasing to us and concluded that the predominant reason for this is that yellow is very dark. If the yellow color is set to a lighter value the whole palette looks a lot better. This is confirmed by plotting all Brewer palettes in one view, which shows that Cynthia Brewer never uses dark yellow colors.

Therefore, we choose not to use colors of equal lightness. This increases the color distance between the different colors drastically, allowing for more colors in the palette because of increased distinguishability. The use of colors of different lightness also makes it possible to use a greater saturation range with less clamping.

Our approach is to draw a line through all hues from light yellow to dark blue and back to light yellow as shown in Fig. 7-5. Compare this to the palette in Fig. 7-4 to see that our method yields more vibrant and colorful palettes. Also the distinguishability between colors is much higher in our palette, of which the complete definition is given on the following page.

Qualitative palette definition

Palette properties:

$\Sigma_{qual} = [1..\infty] \times [0..1] \times [0..1] \times [0..1] \times [0..1] \times [0..1]$

$\sigma_{qual} : \Sigma_{qual}$

$\sigma_{qual} = \langle N, \varepsilon, r, s, b, c \rangle$

Palette:

$P_{qual} : \Sigma_{qual} \times [0..1] \mapsto LCH$

$P_{qual}(\sigma_{qual}, t) = c_{qual}$

where

$c_{qual}.L = (1-\alpha)\ell_0 + \alpha\ell_1,$

$c_{qual}.S = \min\left(\mathrm{S}_{\max}(c_{qual}.L, c_{qual}.H),\ s \cdot s_{max}\right),$

$c_{qual}.H = 360\,(\varepsilon + tr) \bmod 360.$

$\alpha = \mathrm{HueDiff}(h,\ c_y.H)$

$$\mathrm{HueDiff}(h_0, h_1) = \begin{cases} |h_1' - h_0'|, & |h_1' - h_0'| < 180 \\ 360 - |h_1' - h_0'|, & \text{otherwise} \end{cases},$$

where $h_0' = h_0 \bmod 360, h_1' = h_1 \bmod 360.$

$s_{\max} = LCH(RGB(1,0,0)).S$ // The maximum saturation possible

$\ell_0 = b \cdot c_y.L$

$\ell_1 = (1-c) \cdot \ell_0$

$c_y = LCH(RGB(1,1,0))$ // yellow

Discrete palette:

$P_{qual} : \Sigma_{qual} \times [0,1..N) \mapsto LCH$

$P_{qual}(\sigma_{qual})[i] = P_{qual}\left(\sigma_{qual}, \dfrac{i}{N}\right),\ i \in [0,1..N)$

Default values:

$\varepsilon = 0$

$r = 1$

$s = 0.5$

$b = 1$

$c = 0.5$

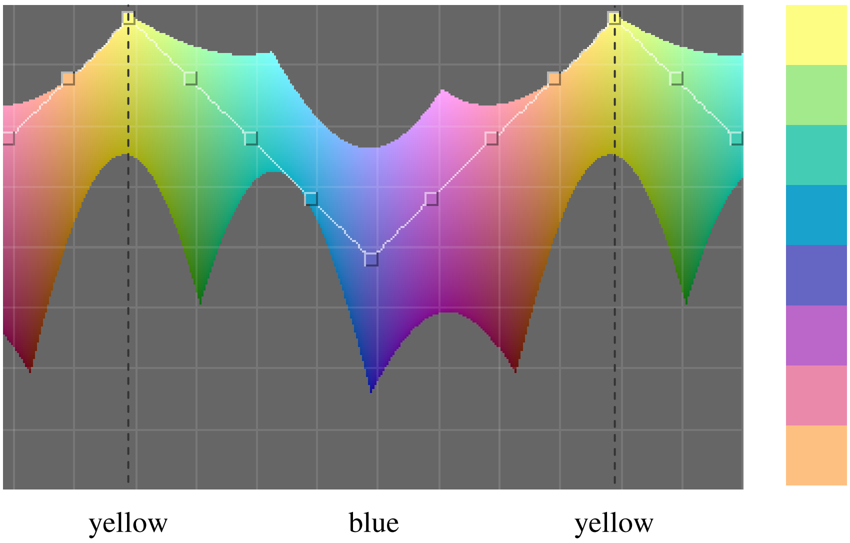


Fig. 7-5 Saturation slice of S = 80. The qualitative palette follows a line from light yellow to dark blue to light yellow.

The palette properties Σ_{qual} contain *s, b* and *c*¸ which respectively stand for saturation, brightness and contrast, just as with the other generated palettes. Two new parameters are added to maintain a flexible model. The *r* parameter represents the used hue range. The ε represents the hue shift, which controls where the range of hues begins.

This leads to the following definition of the hue

$$c_{qual}.H = 360\,(\varepsilon + tr) \bmod 360.$$

The lightness is then determined using the hue distance to yellow

$$c_{qual}.L = (1-\alpha)\ell_0 + \alpha\ell_1,$$

where

$$\begin{aligned}
&\alpha = \mathrm{HueDiff}(h,\ c_y.H),\\
&\ell_0 = b \cdot c_{yellow}.L,\\
&\ell_1 = (1-c)\cdot \ell_0,\\
&c_{yellow} = LCH(RGB(1,1,0)).
\end{aligned}$$

ℓ_0 is the lightness of the lightest color, ℓ_1 is the lightness of the darkest color. The brightness controls how light ℓ_0 is. The contrast determines the lightness difference between ℓ_1 and ℓ_0. HueDiff is a function that calculates the difference between two hues. The difference between *h* and the hue of yellow determines what the lightness of the color is.

The saturation is

$$c_{qual}.S = \min\left(\mathrm{S}_{\max}(c_{qual}.L, c_{qual}.H),\ \ s\cdot s_{max}\right),$$

where

$$s_{max} = LCH(RGB(1,0,0)).S \quad \text{// The maximum saturation possible}$$

In CIELUV, the red hue has the most saturated MSC of all hues. Therefore *s* scales the saturation from 0 to s_{max}. The minimum of $\mathrm{S}_{\max}(c_{qual}.L, c_{qual}.H)$ and $s\cdot s_{max}$ clamps the color back to the triangular approximation surface, which minimizes errors resulting from clamping.

By using this definition of P_{qual} a wide range of qualitative palettes can be generated as can be seen in Fig. 7-7.

Generated qualitative palette evaluation:

- **Distinguishability:** The distinguishability is generally lower than with the fixed-color palette; the difference gets larger as the number of colors is increased. However, distinguishability is a lot higher than the generated qualitative palettes of Healey or Zeileis and Hornik.
- **No ordering:** The palette does show an ordering, although this might be exaggerated by the way the palette is presented. For fixed-color palettes, the colors can be ordered such that there seems to be no clear ordering. If the colors of the generated palette are reordered then the ordering is not so clear (Fig. 7-6).
- **Equal-importance:** Because all colors have equal saturation no color stands out between the rest; therefore, the generated palette satisfies the equal-importance requirement.
- **Nameability:** Since the generation method does not consider nameability, the colors have no clearly defined names.

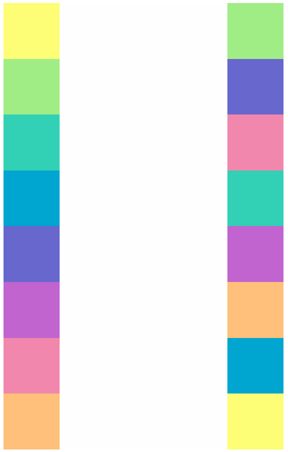


Fig. 7-6 The left palette seems much more ordered than the right one although the colors are exactly the same.

The great advantage of these palettes is that the brightness, saturation and contrast can be altered. And, because all colors have equal saturation, they all seem equally important. Furthermore, we find that the palettes generated with P_{qual} look very balanced and in harmony, creating pleasantly looking palettes. It is hard to generate an ugly looking palette using this method. Finally, this technique enables generation of palettes with any number of colors. But the higher the *N* the more ordering is seen and the lower the distinguishability is.

Our opinion is that the generated palettes generally look better than the fixed-color palettes. It does not matter whether the colors are used in order or randomly shuffled, they always look good. However, from a theoretical point of view, the fixed-color palettes usually satisfy the more important requirements on qualitative palettes. Therefore, our recommendation is to use a fixed-color palette when nameability, distinguishability and no ordering are desired. If equal importance or adaptability is desired, to make the palette meet visualization-specific requirements, then a generated palette can be considered.

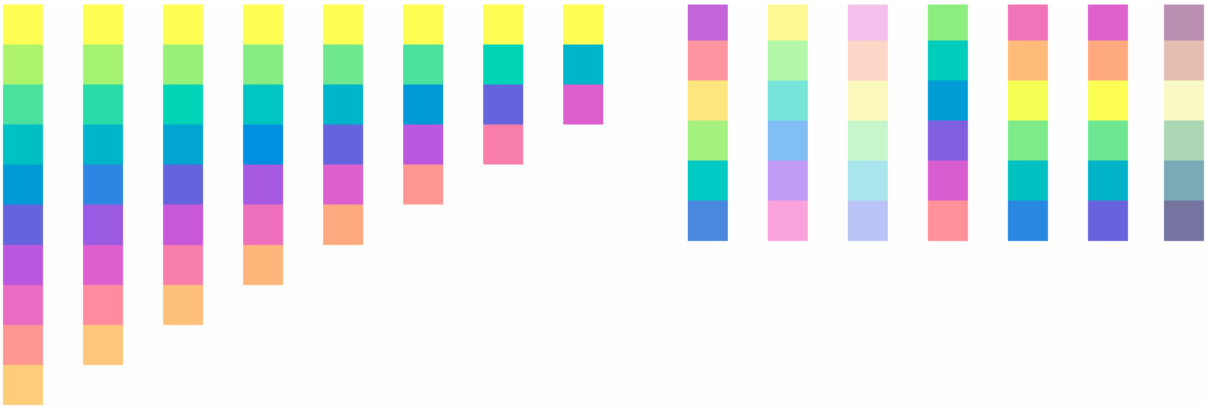


Fig. 7-7 The left eight palettes have the default settings. The right group of palettes show some of the variations possible with this definition.

8 Application

The generation techniques described in this thesis enable users to create a wide range of palettes, using parameters rather than color keys. The quantitative palettes generated with the presented methods meet the palette requirements, for most parameter settings. Exceptions to these are palettes having a contrast of 0, then there is no ordering; and if saturation is very high, e.g., $s > 0.9$, not all colors are perceived equally important.

To show that this palette generation technique enables the creation of proper palettes, the Brewer palettes are paired to generated palettes in Fig. 8-1. By tuning the parameters, it is possible to create generated palettes that closely resemble the Brewer palettes. We only compare the sequential and diverging palettes, since they have more general appearances, whereas the qualitative palettes from Brewer do not have such uniform appearances, as discussed in section 7 Qualitative palettes.

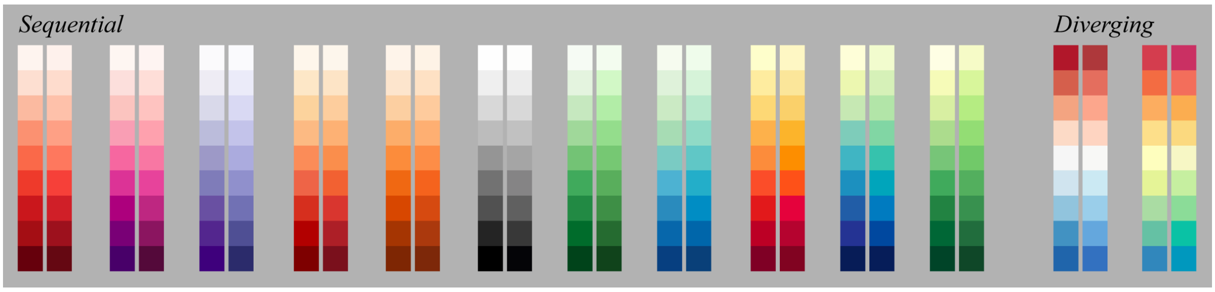


Fig. 8-1 Pairs of palettes, each left palette is a Brewer palette, each right palette is generated and tuned to closely resemble the corresponding Brewer palette.

In the tuned palettes, the contrast and brightness parameter are relatively close to their proposed defaults, in ranges [0.78..1] and [0.76..1] respectively. An interesting detail is that all single-hue palettes, except for the Purples, had a cold/warm value in range of [0.15..0.3]. If w was set to 0 for these palettes, they look dull and cold compared to the Brewer single-hue palettes.

The saturation parameter varies from 0.45 to 1, if the Grays palette is omitted. The reason for the high variation in saturation is twofold. Firstly, there is a gamut difference between the Brewer palettes and the generated palettes; the Brewer palettes are designed for printed display and the generated for emissive displays. In the RGB gamut, green and purple can be more saturated than in CMYK. Therefore, the corresponding Brewer palettes are relatively unsaturated within the RGB space. To recreate them, a low s value has to be chosen. The second cause for high variation in the s parameter is explained by our perception of yellow, orange and red colors. If colors of these hues are lowly saturated, they are perceived as brown. Therefore, if we create an ‘orange’ palette with $s = 0.6$, it looks more brown than orange. To make the palette look orange, the saturation has to be increased. This does not necessarily yield a better palette, since it accentuates the very saturated colors. Fig. 8-2 shows a number of quantitative palettes generated with the presented generation methods.

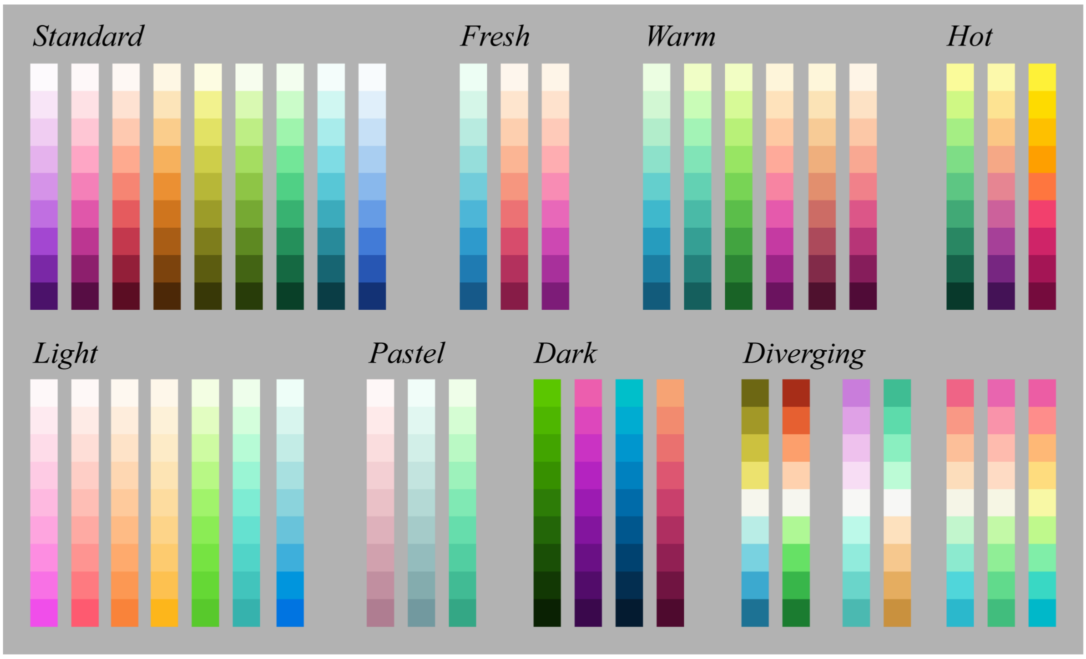


Fig. 8-2 Examples of generated quantitative palettes, which shows the wide range of palettes that can be created. These palettes are a subset of the set of presets used in MagnaView. Choosing from presets enables users to quickly create a palette.

8.1 Integration in MagnaView

The palette generation techniques from this thesis are integrated in the visualization tool MagnaView [17]. This tool visualizes multivariate business data using generalized treemaps [29], seen in Fig. 8-4. Each tile in the visualization is assigned a color according to a mapping from data values of a single attribute to colors. It is possible in MagnaView to create cushioned treemaps (Fig. 8-3); then, the tiles are given a shading based on an RGB shading algorithm.

The multivariate data, which MagnaView visualizes, are divided in attributes. For example, if the telephone usage over a number of years is used as input data, attributes might be: person, telephone number, duration, costs, time of calling, etc. This shows that attributes can have various data-types: integer, real number, string, date or time.

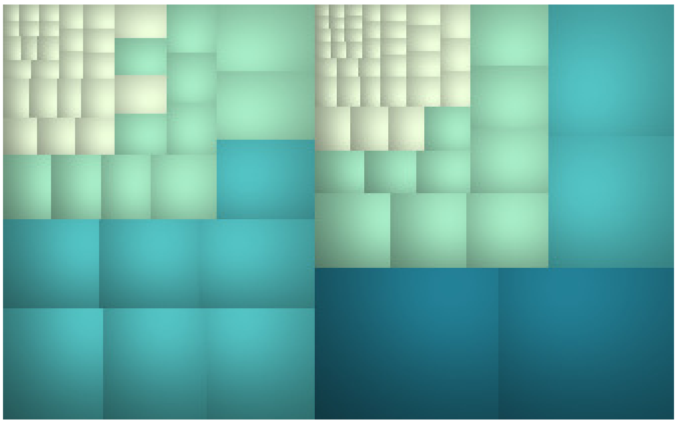


Fig. 8-3 Cushioned tiles in a squarified treemap.

There are several types of attributes: input data, expressions and maps.

- Input data are 'normal' attributes, such as person, telephone number and duration in the telephone usage example above.
- Expressions are defined by giving a formula, possibly using other attributes, examples are "cost * duration / 60", or "cost / sum_all(cost)".
- Maps are created by categorizing the values of a data store or expression in a number of ranges. An example would be to categorize the costs of the calls in the ranges "less than € 1.00", "€1.00 to €5.00" and "more than € 5.00".

8.1.1 Former situation

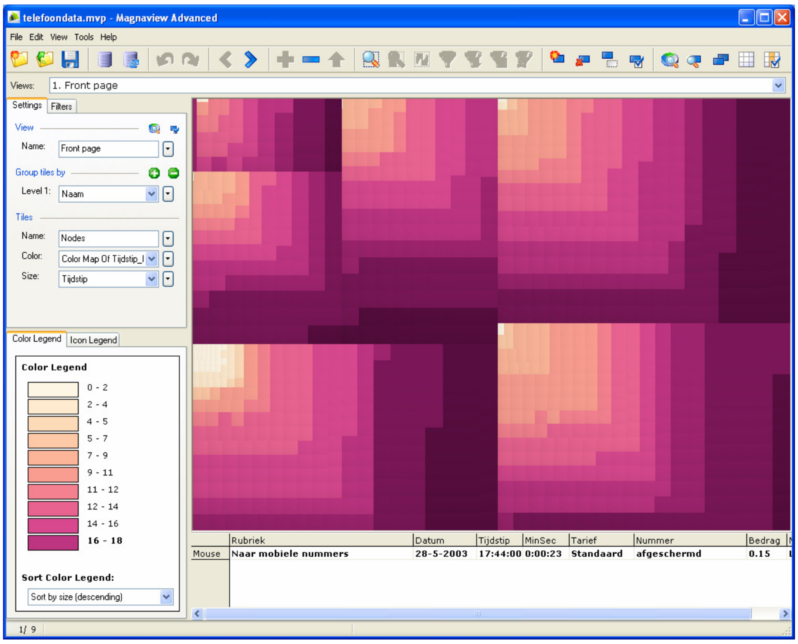
The former palette generation algorithm in MagnaView used the HLS color space. Based on two color keys, a path through the HLS space was obtained by linearly interpolating the HLS coordinates. Since the intermediary hues can take a short or a long path along the color circle, users could choose either "colorize short" or "colorize long".

The attributes were assigned default palettes and colors could only be changed for maps. If the data-type was string, then a qualitative palette was generated by using white and yellow with 'colorize long'. This yields colors of many different hues and varying lightness. For the other data-types, a quantitative palette was obtained, by using white and red with 'colorize short', yielding a white to red palette. For the map attributes a new palette could be generated and each color could be changed individually.

8.1.2 New situation

In the new situation, the proposed palette generation techniques are integrated in MagnaView.
The attributes are assigned a palette type based on their data type:

- Strings are assigned a qualitative palette.
- Unsigned numerical, date and time data are assigned a sequential palette.
- Signed numerical data is assigned a diverging palette.



Views: 1. Front page

Name: Front page

Level 1: Naam

Name: Nodes

Color: Color Map Of Tijdstip_l

Size: Tijdstip

Color Legend

▢	0 - 2
▢	2 - 4
▢	4 - 5
▢	5 - 7
▢	7 - 9
▢	9 - 11
▢	11 - 12
▢	12 - 14
▢	14 - 16
▢	**16 - 18**

Sort Color Legend: Sort by size (descending)

	Rubriek	Datum	Tijdstip	MinSec	Tarief	Nummer	Bedrag
Mouse	**Naar mobiele nummers**	**28-5-2003**	**17:44:00**	**0:00:23**	**Standaard**	**afgeschermd**	**0.15**

Fig. 8-4 The MagnaView application. The colors on the tiles are from a generated palette.

The data store and expression attributes are assigned palettes that can be edited, whereas in the old situation these attributes had fixed colors. The sequential and diverging palettes are continuous palettes, since no default categorization is defined on these data types. Qualitative palettes are always discrete palettes.

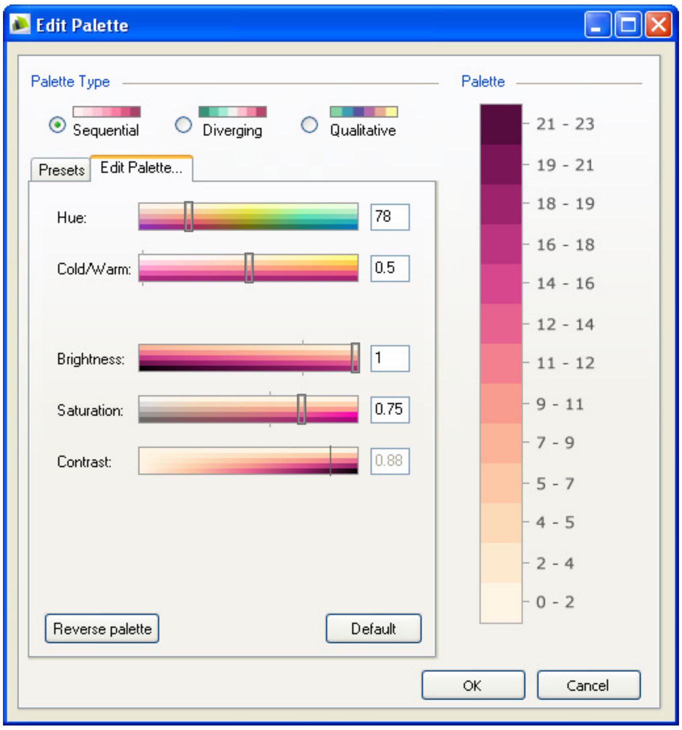
The map attributes are assigned discrete palettes, since they always have a finite number of data values, defined by a given categorization.

8.1.3 User interface

The user interface for editing palettes is layered, in order to provide easy input for several types of users:

Novel users can choose a palette from a set of preset palettes, of which a subset is shown in Fig. 8-2. Some palettes have a variable contrast value depending on the number of colors, using the relation described in section 5.7; some palettes have a fixed contrast value, so the palette looks the same for all *N*. That way a preset can be created, for example, that ranges from white to the most saturated red. The saturated red does not change lightness if the number of colors is changed. Novel users might sometimes have to change the palette type, if the default palette type was not the desired one; for instance, sometimes numbers represent qualitative data.

Experienced users can change the palette parameters using sliders (Fig. 8-5). Each slider shows how the palette changes, if the corresponding parameter is altered. This way, the user can see beforehand how changing a certain parameter will impact the appearance of the palette. When one parameter is altered, the other slider appearances change accordingly. Since the preset palettes are generated using the same parameters, it is possible to select a preset palette and then alter that palette, using the slider interface.



Edit Palette

Palette Type: ● Sequential ○ Diverging ○ Qualitative

Hue: 78

Cold/Warm: 0.5

Brightness: 1

Saturation: 0.75

Contrast: 0.88

Fig. 8-5 The user interface used to edit palettes in MagnaView. This palette is used in the visualization in Fig. 8-4.

Expert users can still alter each color individually for map attributes. This can only be done for map attributes, since these always have discrete palettes. If, for instance, a certain range of numerical data needs to be colored red, one must first define a categorization with a map attribute, then the appropriate range can be set to red.

The user interface for editing palettes is created in a separate version of MagnaView that is not in use. It will, however, be integrated in the next release version of MagnaView. This means that we have no information on how the user interface is used in practice. To get more insight in how the palette editing interface would function we performed a user study described in the next section.

8.2 User study

The objective of this user study is to investigate how test subjects handled the user interface for editing palettes and to see which palettes were judged to be most aesthetically pleasing. The user study was performed with 15 non-colorblind subjects, of which seven were MagnaView employees, who had experience with the MagnaView application, and eight people who had never before used MagnaView. The user study consisted of three successive parts, which are described in the subsections below.

8.2.1 User interface test

Each subject was asked to perform a set of 6 increasingly difficult tasks; ranging from selecting a green palette to creating a palette that looks nice for a given view. A MagnaView project with 6 different views was used for this test. Each view showed the corresponding task description. If a subject finished the task they were allowed to continue to the next view. These tasks were intended to familiarize the subjects with the palette editing user interface. We evaluated the subjects during this test and questioned them afterwards.

The fact that the interface was 'layered' received a positive response from most test subjects. The selection of a preset to obtain a palette was clear. To specify a palette other than the presets, the sliders seen in Fig. 8-5 need to be used. Some subjects did not know the meaning of all palette property terms, such as 'hue' and 'saturation'. However, all proceeded by experimenting with the sliders. After a short learning period of about 5 to 10 minutes, most subjects were able to understand the workings of all sliders and could adapt the palettes in a way they wanted. However, some operations require the change of more than one parameter; for example, when the darkest color of a palette with high contrast needs to be brightened. Then, the contrast needs to be lowered and the brightness set to a high enough value. This was often perceived as a difficult task, which some users were unable to solve. Some users asked if it was possible to change the top or bottom color in order to perform this operation. Furthermore, for a small set of parameter settings, the change of certain parameters does not change the palette appearance; for instance, if the contrast and brightness are low, changing the cold/warm parameter seems to have no influence on the palette. This was confusing to new users.

From the MagnaView part of the testers, the new coloring system received a positive response. Many of them never used the old coloring algorithm or selected colors manually, often resulting in weird color combinations. Therefore, the selection of colors was often left to the designer who picked appropriate colors by hand. The new system makes it possible for everyone to create aesthetically pleasing palettes, although the designer can still handpick each color when special palettes are required. We expect that the average MagnaView user will use the preset palettes most of the times.

Our overall conclusion is that people are able to generate palettes fairly easy after a short learning period using this interface. The fact that the interface is layered guides the users in the learning process, and we suspect that for most of the palette uses selecting presets will suffice. Even though the parameter names are not always understood properly, the sliders provide enough visual clues to guide the user towards the correct actions.

8.2.2 Qualitative palette methods

In the second part of the user study, we tested the aesthetic properties of the two methods for generating qualitative palettes described in section 7.1 and 7.2. The test subjects were shown eight PowerPoint slides. Each slide showed a different MagnaView view, which was colored using a generated palette and using a fixed-color palette (Fig. 8-6). The subjects were asked which view they liked best, A or B, and to judge only based on appearance.

Table 8-1 shows, for each slide, the percentage of testers that voted for the generated color palette and the percentage that voted for the fixed-color palette. The table shows that the generated palettes are generally judged to be most aesthetically pleasing. One of the reasons for this is that the generated palette has more subtle changes between adjoining colors. Users commented during the study that they found the method A palettes more peaceful and less aggressive than the method B palettes. Another cause for the dislike of the fixed-color palettes seems to be the presence of gray and brown colors, of which some subjects showed their dislike.

There were, however, views in which the fixed-color palette scored best. We found two reasons for this. Firstly, if the data values are not distributed evenly among the different categories, then the generated palette can look out of balance. It could, for example, lead to a view with much lime green and a number of reddish pink colors; a combination that most subjects disliked. The fixed-color palettes suffer less from this, since they have less harmony to begin with. Secondly, when the colors are surrounded by large portions of neutral space, their appearance is more tranquil. In that case, the fixed-color palette is perceived as less chaotic and less aggressive than usual, which can lead to a better looking visual.

In conclusion, although we know that the fixed-color palettes are theoretically superior to the generated palettes, this study shows that in most situations the generated palettes yield a more balanced visualization, which is liked by most test subjects. However, in some specific cases the fixed-color palette can yield a more aesthetically pleasing result.

Slide nr.	Generated palettes	Fixed-color palettes
1	33%	67%
2	80%	20%
3	40%	60%
4	87%	13%
5	73%	27%
6	93%	7%
7	27%	73%
8	60%	40%
Total	**66%**	**34%**

Table 8-1 The percentages of test subjects that voted for a generated or fixed-color palette.

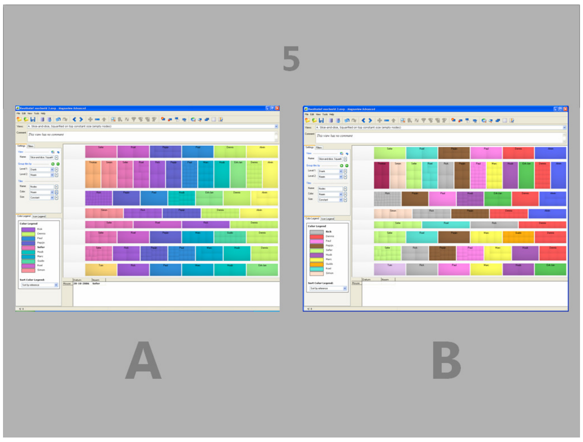


Fig. 8-6 Slide 5 of part 2 of the user study. The left one shows a view with generated qualitative colors, and the right one has fixed colors

8.2.3 Sequential palettes

In the final part of the user study, the aesthetic qualities of the sequential palettes were investigated. On the one hand this was done to compare the generated palettes to the Brewer palettes, on the other hand we wanted to use this information to choose default palettes for MagnaView. We created a view that was colored with 48 different palettes divided in 10 categories, which are shown in Table 8-2. All categories with generated palettes contain five palettes; the Brewer categories contain the number of palettes available for that category. Screenshots of these views were put in a PowerPoint presentation. Three versions of the presentation were made, in which the slides were shuffled differently to avoid order effects. Also an example presentation was made with seven views that was intended to represent the complete population.

The subjects were shown the seven example slides first, to give them an idea of what they could expect. The users were given a form on which they could rate each view on a scale from 1 to 5, from 'ugly' to 'beautiful'. Then one of the three presentations with 48 views was shown to each test subject. The users were asked to only pay attention to appearance, not to functionality of the palettes.

Code	Type		Average	Std. Dev.
BS	Brewer	Single-hue	**3,44**	0,33
G1	Generated	Multi-hue (w = 0.25)	**3,40**	0,40
BY	Brewer	Multi-hue (Yellow-topped)	**3,09**	0,42
GC	Generated	Contrast low	**3,08**	0,31
GS	Generated	Single-hue	**3,03**	0,51
BP	Brewer	Multi-hue (Pink-topped)	**2,93**	0,74
G2	Generated	Multi-hue (w = 0.6)	**2,93**	0,38
BC	Brewer	Multi-hue (Cyan-topped)	**2,70**	0,09
G3	Generated	Multi-hue (w = 1)	**2,57**	0,25
GD	Generated	Diverging	**1,96**	0,25

Table 8-2 The average ratings of the palettes of each category.

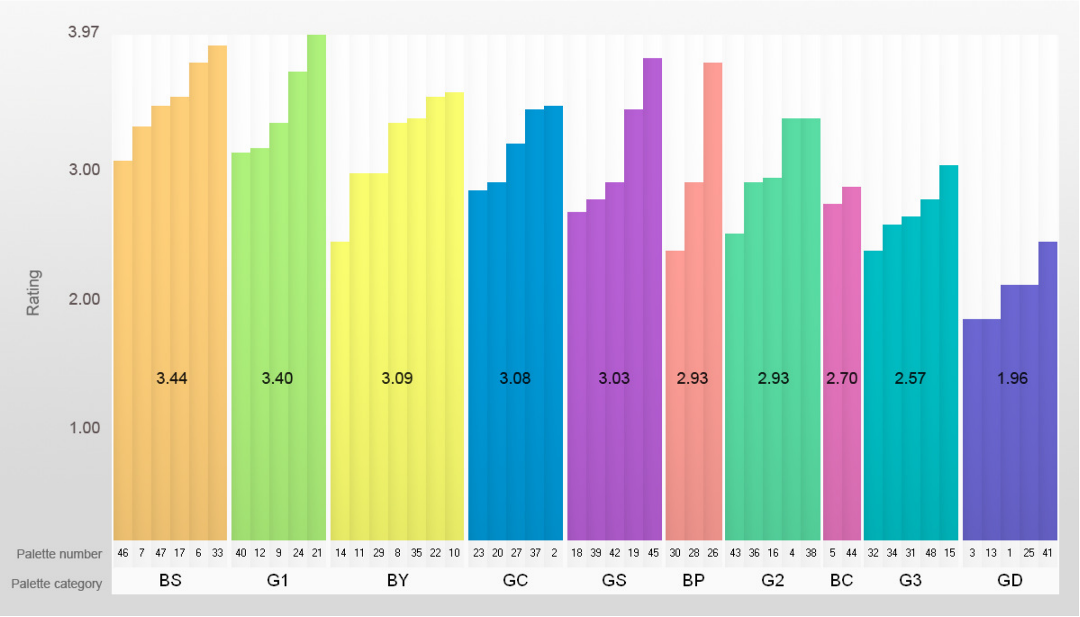


Fig. 8-7 A visualization made using MagnaView. Each bar represents the rating of one of the palettes in the sequential palette user study. (Note that the bars are colored using generated qualitative colors)

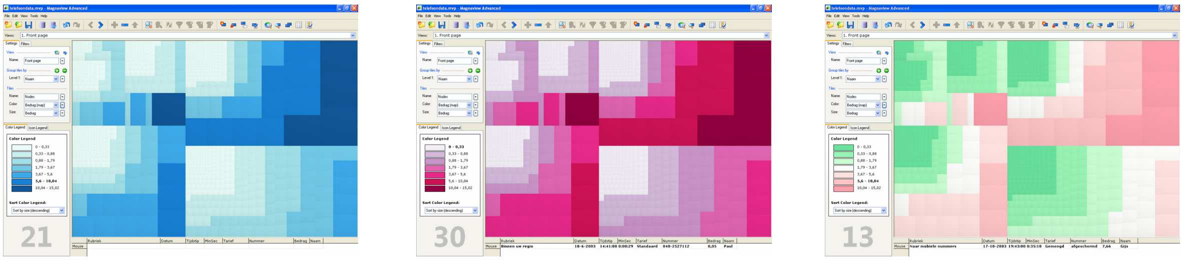


Fig. 8-8 From left to right: the highest rated palette, the lowest rated sequential palette, the lowest rated palette.

Color	Number of palettes	Average
Blue	4	**3,75**
Green	3	**3,37**
Cyan	4	**3,31**
Red	3	**3,09**
Pink	3	**3,07**
Orange	2	**2,95**
Yellow	1	**2,57**

Table 8-3 The single-hue sequential palettes rated by hue. Only the palettes with a low hue range were used, i.e., palettes from the BS, GS, G1 and GC category.

Table 8-2 shows the average rating of palettes of each category. In Fig. 8-7 the average ratings of each palette are shown by bars, grouped by palette category. Table 8-3 shows the average ratings of the single-hue sequential palettes, grouped per hue. Based on these results a number of observations can be made:

- The single-hue palettes of Brewer scored highest in the test, closely followed by the G1 category. In the beginning of this chapter, we showed that to recreate the single-hue Brewer palettes, the generated palettes had a cold/warm value in the range [0.1..0.3]; therefore, it was to be expected that these two categories would have about the same score.

- Contrary to our expectations, the BS and G1 palettes scored better than the multi-hue palettes of Brewer and generated palettes with a higher cold/warm value. We expected that the users would like a colorful palette ranging over many hues, but this proved not to be the case.

- In section 5.8, we assumed that users would like the yellow-topped palettes better than the pink- and cyan-topped ones. The results confirm this, however, the difference between the yellow- and pink-topped palettes is not very large. In Fig. 8-7, we can see that one palette in the BP category scored much higher than the others; this was a blue palette with a small hue range. If we remove this palette, which looks very much like a single-hue palette, then the average of the BP category is 2.53, yielding a substantial difference with the BY category.

- The diverging palettes scored much lower than all other palettes. We believe the low ratings are a result of the composition of colors in the view. The sequential palettes create an image with fairly smooth transitions, whereas the diverging palette has more abrupt transitions because of the two distinct hues. Therefore, we believe that the sequential palettes do not look that much nicer than the diverging palettes in isolation. However, it is harder to create a balanced and aesthetically pleasing composition using diverging palettes.

- We expected the low contrast palettes from the GC category to score higher than the other palettes. The lower contrast yields a more tranquil and quiet view. However, the rating of the GC palettes did not reflect this.

- The three palettes with the highest rating were all single-hue blue palettes, from the categories G1, BS and GS, with ratings of 3.97, 3.87 and 3.77 respectively. Table 8-3 also shows that the users had a preference for blue palettes. Note that this table only shows the averages of palettes from the BS, GS, G1 and GC categories, since they have a relatively low hue range.

In conclusion, the palettes that received the highest rating were the single-hue Brewer palettes and the generated palettes with a low cold/warm value. The palette with the highest rating was a generated blue palette with a low cold/warm value. Therefore, this palette would be the obvious choice for a default palette in MagnaView.

8.3 Recommendations

Below we give a number of recommendations on how to define palettes:

- If other aspects of the visualization are more important than the tiles (or the surfaces for which the palette colors are intended), then a lowly saturated palette with low contrast is recommended. That way a more tranquil picture is obtained.

- If segmentation is important, i.e., the detection of which color belongs to which value, then the palette should have a high enough contrast value. Increasing the saturation and cold/warm parameter also add to the detectability, although to a lower degree than contrast. The higher the spatial frequency, the higher the contrast, saturation and cold/warm need to be, to have proper detectability.
  If the cushion height is high, then shading is darker. To retain detectability, contrast should be increased when cushion height is increased. The increase of saturation and cold/warm also helps, but has lower impact.

- For sequential palettes, if saturation is set to a high value, some colors might be perceived as being more important. This can be used to emphasize certain regions that demand more attention. However, if this is not desired, the brightness and contrast can be adjusted to set the most saturated color of the palette to the begin or end color of the palette. This way, an ordering in saturation is perceived and the palette requirements are satisfied. Examples of this are showed in Fig. 8-2, the “Light” palettes are created this way.

- For sequential palettes, if palettes need to be generated that are suitable for colorblind persons, multi-hue palettes are not recommended. The luminance perception of colorblind people differs from not colorblind people, therefore palettes with a high *w* value might have no clear ordering for a colorblind person. Reducing the saturation decreases this effect.

- For diverging palettes, hue combinations suitable for colorblind people are: red-blue, red-purple, orange-blue, orange-purple, brown-blue, brown-purple, yellow-blue, yellow-purple [4].

- For diverging palettes, the hue difference should be high enough to be able to differentiate between below or above midpoint. We recommend a minimal hue difference of 80, preferably higher. If spatial frequency is high, saturation is low, or cold/warm is high, hue difference should be increased.

- For qualitative palettes, we recommend to use a fixed-color palette, if segmentation is important. If not, the generated palette generally yields more tranquil colors combinations.

9 Conclusion

In this thesis we presented models to generate sequential, diverging and qualitative palettes. The requirements on these palettes are specified in terms of the perceptual properties hue, saturation and lightness. In order to generate palettes that satisfy these requirements, a suitable color system is needed. Due to the perceptual errors in the widely used HSV and HLS color spaces, the perceptually more uniform CIELUV color space was used as a basis. However, CIELUV is a relative color system, which means that not all CIELUV colors have a corresponding RGB color. Therefore, it is important to be able to define where the boundaries of displayable color space lie, in order to use the color system to automatically generate a palette.

The CIELUV color space is inherently more complex than HSV or HLS. Since we found few insightful pictures of the CIELUV color space in books and on the internet, we developed the PaletteView application to analyze the CIELUV color space. This application enables fast exploration of color spaces by looking at slices. The four viewports in PaletteView can be used to compare several views of a single color space or to compare different color spaces with each other. The intersections provide an easy means to analyze a 3D volume of colors.

By clicking on the slices in PaletteView a set of colors can be selected to form a palette. Each color can be moved around on a chosen intersection plane to edit the palette. If a color is moved in one viewport, the position of that color in the other viewports changes accordingly. Therefore, PaletteView not only allows for easy analysis of various color spaces, it also makes it possible to see the relations between different color spaces; a set of colors can be viewed within different slices in different color spaces at the same time. Furthermore, PaletteView is well suited to use as a research tool to quickly assess the quality of new palettes or methods of palette generation by clicking together a set of colors. Also, PaletteView can be used to analyze existing palettes or color combinations within different color spaces. We did this with the Brewer palettes, which we used as an inspiration to generate our palettes.

Using PaletteView, we were able to simplify the boundaries of displayable CIELUV color space to a triangular surface for each hue. This involved the derivation of the Most Saturated Color for a hue; the MSC(h). The triangular surface was used as a convex hull for the sequential palette curve. We created a flexible curve, of which the tension can be altered, by combining two normal Bézier curves. By adjusting the tension of the curve the saturation of the palette can be varied. The range on the curve was restricted to control the brightness and the lightness contrast of the palette. Based on the Brewer palettes, we defined a linear function that approximates the default contrast value for a palette with a given number of colors. The cold/warm parameter controls the hue range of the palette; the higher the value is, the closer the top point of the curve is to yellow. The addition of the cold/warm parameter makes it possible to generate multi-hue palettes in a convenient way.

To generate a discrete palette the curve is sampled. A Bézier curve, and therefore also our compound curve, is not arc-length parameterized. Therefore, the position of a point on the curve is calculated using a linear piecewise approximation. Although CIELUV is more perceptually uniform than HSV and HLS, it is not perfectly perceptually uniform. This shows clearly when the curve is sampled at regular intervals; uniform lightness increases were not perceptually uniform. The CIE has tried to compensate for the perceptual errors in the original CIELUV space by defining new distance functions. These, however, posed no significant improvement on the original CIELUV space for this application. Therefore, we designed a distance function that transforms the lightness axis of the CIELUV space, based on the lightness increases of the Brewer palettes. The sequential palettes that use this new function look more uniform than their counterparts that use the CIE distance functions.

The sequential palette definition was then used to create a diverging palette model. Two sequential palettes are joined at the lightest color to create a diverging palette; the lightest color is replaced by a neutral color. To make the diverging palette more practically usable, we made the midpoint position variable. When the midpoint is moved, the palette is translated and scaled in order to satisfy the requirements for diverging palettes. For a discrete diverging palette the midpoint position determines whether the palette contains a neutral color or not. This model for diverging palettes allows users to set the midpoint position to a custom range, such as the mean, or zero-point of a set of values. It also allows them to choose whether the palette contains a neutral midpoint color or not.

Although we mainly focused on quantitative palettes, we also developed a generation technique for qualitative palettes. The technique is based on a path through all hues. The path is defined by the parameters brightness, saturation, contrast, hue range and the begin hue. The colors are clamped using the triangular surface that approximates the displayable colors for each hue. Contrary to other generation methods, we allow our path to change in lightness, which yields more distinguishable colors in the palette. However, the fixed 17-color palette we designed satisfies more of the given requirements than the generated palette. Therefore, for most applications that require up to 17 colors, the fixed-color palette is suited best.

The models to generate sequential, diverging and qualitative palettes were implemented in the visualization tool MagnaView. For numerical data continuous palettes can be used, or the data can be categorized for use with a discrete palette. The user interface to edit palettes was split up in three levels: novel users can choose from a set of preset palettes, experienced users can change the properties of a palette via sliders, or expert users can change each color individually. This allows for a broad range of interaction with palettes; from very fast palette selection to very detailed.

We performed a user study to test the user interface in practice. This showed that inexperienced users were able to use the sliders to create palettes of their liking after a short learning period. However, some operations require the change of multiple sliders, which was perceived as a difficult task. The user study was also intended to investigate the aesthetic qualities of the palettes. The generated qualitative palettes were judged to be more aesthetically pleasing than the fixed-color palettes in most situations. Concerning the sequential palettes, we found that the users liked the single-hue palettes better than the palettes that used a large hue range. The Brewer single-hue palettes and generated palettes with a cold/warm value of 0.25 scored highest in the test. The test subjects showed a preference for blue single-hue palettes.

In conclusion, the PaletteView application allows for a new method of color space and palette analysis. It allows for easy analysis and comparison of various aspects of color spaces, which are hard to grasp just by looking at the formulae. Furthermore, the insight and information gained by using PaletteView allowed us to specify palettes in a new way. Instead of giving vague guidelines on how palettes should be generated, we were able to specify explicitly how the palettes are generated. Furthermore, the palettes are not specified using color keys, but using properties of palettes as parameters. This leads to a higher level of palette specification, in which aspects of a palette can be altered. The set of possible palettes is thus restricted to a set that satisfies the palette requirements. By changing these properties a palette can easily be tuned to meet visualization-specific requirements.

9.1 Future work

The method used to create multi-hue sequential palettes, shifts the top point of the palette towards a given bright point. Since the bright point is not at lightness 100, the lightness of the palette is slightly decreased as cold/warm is increased. We have considered two multi-hue methods that did not suffer from this, however, one resulted in dull unsaturated palettes, the other resulted in palettes with irregular saturation. Further research on this subject might lead to a method that retains the colorfulness of the current multi-hue palettes, while using the complete lightness range.

The presented palette generation techniques use properties to specify palettes in an exact way. These techniques can be used to automatically generate palettes in a similar way to PRAVDAColor [2]. Based on certain aspects of the data and the visualization, such as data distribution and spatial frequency, a palette specification in terms of palette properties can be derived. Based on user input, this specification could be changed to fit the intended task: segmenting, highlighting, showing outliers, etc. The derived specification in terms of palette properties can be fed to the proposed palette generation model, to automatically obtain a suitable palette. This leads to a higher level of palette generation. The fact that our methods use palette properties to specify palettes makes it especially suited for this kind of use.

The PaletteView application is a tool that can be used for further color space or palette research. New color spaces and the set of accompanying parameters can be explored and compared to other color spaces using PaletteView. Furthermore, new methods of palette generation can be quickly tested and analyzed within the context of the color spaces that are currently integrated in PaletteView.

It is conceivable that the viewports of PaletteView are integrated in the palette editing user interface. This way, users can see the path of the palette through color space. By dragging the colors around on slices, the palette colors can be changed. This might be a convenient method for expert users to edit their palettes.

In this thesis, we have considered the problem of generating a color palette. In visualizations this color palette is always used in combination with data values. The development of a well-designed user interface to specify a mapping, or categorization, between data values and colors is suggested for future research.

Appendix

Color transformations

RGB to HSV

HSV is also called HSB. The transformation from RGB coordinates to HSV coordinates is as follows:

$$MIN = \min(R,G,B)$$

$$MAX = \max(R,G,B)$$

$$\delta = MAX - MIN$$

$$H = \begin{cases} 60\left(0+\frac{G-B}{\delta}\right), & R = MAX \\ 60\left(2+\frac{B-R}{\delta}\right), & G = MAX \\ 60\left(4+\frac{R-G}{\delta}\right), & B = MAX \end{cases}$$

$$S = \delta$$

$$V = MAX$$

RGB to HLS

HLS is also called HSL or HSI. The transformation from RGB coordinates to HLS coordinates is as follows:

$$MIN = \min(R,G,B)$$

$$MAX = \max(R,G,B)$$

$$\delta = MAX - MIN$$

$$H = \begin{cases} 60\left(0+\frac{G-B}{\delta}\right), & R = MAX \\ 60\left(2+\frac{B-R}{\delta}\right), & G = MAX \\ 60\left(4+\frac{R-G}{\delta}\right), & B = MAX \end{cases}$$

$$S = \begin{cases} \frac{\delta}{2L}, & L \le 0.5 \\ \frac{\delta}{2-2L}, & L > 0.5 \end{cases}$$

$$L = 0.5\,(MIN + MAX)$$

RGB to XYZ

From a given RGB coordinate in $[0..1]^3$ a set of XYZ $\in [0..1]^3$ values can be calculated. For this transformation a gamma value γ and RGB-matrix M are needed. There are many RGB-matrices, of which a number is given on the website of Bruce Lindbloom [15]. Each RGB-matrix has a corresponding gamma value. We prefer to use Adobe RGB (1998), with gamma value 2.2. For the transformation XYZ to RGB we also refer to the Lindbloom website [15].

$$
\begin{aligned}
&(X,Y,Z)=(r,g,b)\cdot M,\\
&r=R^{\gamma},\\
&g=G^{\gamma},\\
&b=B^{\gamma}.
\end{aligned}
$$

XYZ to LUV and LCH$_{\mathrm{uv}}$

The conversion of XYZ values to LUV requires a reference white. Each RGB matrix uses its corresponding reference white, a table of which can be found on the Bruce Lindbloom website, just as a transformation from LCH$_{\mathrm{uv}}$ to LUV to XYZ [15]. We use D65, of which the coordinates are XYZ(0.95047, 1.000, 1.08883), i.e., $X_n=0.95047,\ Y_n=1.000,\ \ Z_n=1.08883$. The transformation of XYZ to LUV to LCH$_{\mathrm{uv}}$ is as follows.

$$
L^{*}=\begin{cases}116\sqrt[3]{y_n}-16 & y_n>\varepsilon\\ \kappa\cdot y_n & y_n\le\varepsilon\end{cases},
$$

$$
\begin{aligned}
&u^{*}=13L^{*}\left(u'-u'_n\right),\\
&v^{*}=13L^{*}\left(v'-v'_n\right),
\end{aligned}
$$

where

$$
\begin{aligned}
&y_n=\frac{Y}{Y_n},\\
&u'=\frac{4X}{X+15Y+3Z},\\
&v'=\frac{9Y}{X+15Y+3Z},\\
&u'_n=\frac{4X_n}{X_n+15Y_n+3Z_n},\\
&v'_n=\frac{9Y_n}{X_n+15Y_n+3Z_n},
\end{aligned}
$$

$$
\begin{aligned}
&\varepsilon=216/24389\approx 0.008856,\\
&\kappa=24389/27\ \approx 903.3.
\end{aligned}
$$

$$
\begin{aligned}
&C^{*}_{uv}=\sqrt{(u^{*})^{2}+(v^{*})^{2}}\\
&H^{*}_{uv}=\tan^{-1}\left(\frac{v^{*}}{u^{*}}\right)
\end{aligned}
$$

Color distance formulae

CIE 94

The color distance defined by the CIE in 1994 is referred to as the CIE 1994 (ΔL*ΔC*ΔH*) color difference model and is abbreviated as CIE 94, and its symbol is ΔE*94.

The CIE have defined reference conditions under which the new metric with default parameters can be expected to perform well:

1. The specimens are homogeneous in colour.
2. The colour difference (CIELAB) is smaller than 5 units.
3. They are placed in direct edge contact.
4. Each specimen subtends an angle of >4 degrees to the assessor, whose colour vision is normal.
5. They are illuminated at 1000 lux, and viewed against a background of uniform grey, with L* of 50, under illumination simulating D65.

In the following definition the terms *L*, *C*, *H*, *a*, *b*, will be used in stead of *L**, *C_{ab}**, *H_{ab}**, *a**, *b** for increased readability. The difference function to compute the difference between color (L_1, a_1, b_1) and (L_2, a_2, b_2) is defined as

$$\Delta E_{94}^{*} = \sqrt{\left(\frac{\Delta L}{K_L S_L}\right)^2 + \left(\frac{\Delta C}{K_C S_C}\right)^2 + \left(\frac{\Delta H}{K_H S_H}\right)^2} \text{ ,}$$

where

$$\Delta L = L_1 - L_2$$
$$\Delta C = C_1 - C_2$$
$$\Delta H = \sqrt{\Delta a^2 + \Delta b^2 + \Delta C^2}$$
$$C_1 = \sqrt{a_1^{\ 2} + b_1^{\ 2}}$$
$$C_2 = \sqrt{a_2^{\ 2} + b_2^{\ 2}}$$
$$\Delta a = a_1 + a_2$$
$$\Delta b = b_1 + b_2$$
$$S_L = 1$$
$$S_C = 1 + K_1 C$$
$$S_H = 1 + K_2 C$$
$$C = \sqrt{(C_1)^2 + (C_2)^2}$$
$$K_L = 1$$
$$K_C = 1$$
$$K_H = 1$$
$$K_1 = 0.0045$$
$$K_2 = 0.0015.$$

The variables K_L, K_C and K_H are called 'parametric factors' and are included in the formula to allow for adjustments to be made independently to each colour-difference term to account for any deviations from the reference viewing conditions, that cause component specific variations in the visual tolerances. Under the reference conditions explained above, they are set to 1.

CIEDE 2000

The color distance defined by the CIE in 2001 is referred to as the $\Delta E_{00}(L_1^*, a_1^*, b_1^*, L_2^*, a_2^*, b_2^*)$ color difference model and is abbreviated as CIEDE2000, its symbol is ΔE_{00}^{12} or ΔE_{00}.

In the following definition the terms *L*, *C*, *H*, *a*, *b*, will be used in stead of *L**, *C_{ab}**, *H_{ab}**, *a**, *b** for increased readability. The difference function to compute the difference between color (L_1, a_1, b_1) and (L_2, a_2, b_2) is defined as

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{K_L S_L}\right)^2 + \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{K_C S_C}\right)^2 \left(\frac{\Delta H'}{K_H S_H}\right)^2} \ ,$$

where

$$\Delta L' = L_1 - L_2$$
$$\Delta C' = C_2 - C_1$$
$$\Delta H' = 2\sqrt{C_1' C_2'} \sin\left(\frac{\Delta h'}{2}\right)$$
$$C_1 = \sqrt{{a_1}^2 + {b_1}^2}$$
$$C_2 = \sqrt{{a_2}^2 + {b_2}^2}$$

$$\Delta h' = \begin{cases} h_2' - h_1', & |h_2' - h_1'| \le 180° \\ h_2' - h_1' + 360°, & |h_2' - h_1'| > 180°; h_2' \le h_1' \\ h_2' - h_1' - 360°, & |h_2' - h_1'| > 180°; h_2' > h_1' \end{cases}$$

$$h_1' = \begin{cases} \tan^{-1}(b_1 / a_1') & \tan^{-1}(b_1 / a_1') \ge 0 \\ \tan^{-1}(b_1 / a_1') + 360° & \tan^{-1}(b_1 / a_1') < 0 \end{cases}$$

$$h_2' = \begin{cases} \tan^{-1}(b_2 / a_2') & \tan^{-1}(b_2 / a_2') \ge 0 \\ \tan^{-1}(b_2 / a_2') + 360° & \tan^{-1}(b_2 / a_2') < 0 \end{cases}$$

$$a_1' = (1 + G) a_1$$
$$a_2' = (1 + G) a_2$$

$$G = 0.5\left(1 - \sqrt{\frac{\bar{C}^7}{\bar{C}^7 + 25^7}}\right)$$
$$\bar{C} = (C_1 + C_2) / 2$$

$$S_L = 1 + \frac{0.015\left(\bar{L} - 50\right)^2}{\sqrt{20 + \left(\bar{L} - 50\right)^2}}$$
$$S_C = 1 + 0.045\bar{C}'$$
$$S_H = 1 + 0.015\bar{C}'T$$

$$\bar{L} = \left(L_1 + L_2\right)/2$$
$$\bar{C}' = \left(C_1' + C_2'\right)/2$$
$$C_1' = \sqrt{(a_1')^2 + (b_1)^2}$$
$$C_2' = \sqrt{(a_2')^2 + (b_2)^2}$$
$$\begin{aligned} T = \; & 1 - 0.17\cos(\bar{h}' + 30°) + 0.24\cos(2\bar{h}') + \\ & 0.32\cos(3\bar{h}' + 6°) - 0.20\cos(4\bar{h}' - 63°) \end{aligned}$$

$$\bar{h}' = \begin{cases} (h_1' + h_2')/2 & \left|h_1' - h_2'\right| \le 180°; C_1'C_2' \ne 0 \\ (h_1' + h_2' + 360°)/2 & \left|h_1' - h_2'\right| > 180°; (h_1' + h_2') < 360°; C_1'C_2' \ne 0 \\ (h_1' + h_2' - 360°)/2 & \left|h_1' - h_2'\right| > 180°; (h_1' + h_2') \ge 360°; C_1'C_2' \ne 0 \\ (h_1' + h_2') & C_1'C_2' \ne 0 \end{cases}$$

$$R_T = -\sin(2\Delta\theta)R_C$$
$$R_C = 2\sqrt{\frac{(\bar{C}')^7}{(\bar{C}')^7 + 25^7}}$$
$$\Delta\theta = 30\exp\left\{-\left(\frac{\bar{h}' - 275°}{25}\right)^2\right\}$$

The parametric weighting factors K_L, K_C and K_H are usually set to 1. For more information on how to implement these formulae and we refer to the paper “The CIEDE2000 Color-Difference Formula Implementation Notes, Supplementary Test Data, and Mathematical Observations”. This paper also contains test data, which can be used to validate the implementation of the CIEDE2000 distance function.

References

[1] Timo Autiokari , “Evaluation of the the CIE Color Difference Formulas”, http://www.aim-dtp.net/aim/evaluation/cie_de/, 2007

[2] Lawrence D. Bergman, Bernice E. Rogowitz , Lloyd A. Treinish, “A Rule-based Tool for Assisting Colormap Selection”, Proceedings of the IEEE Visualization Conference, pp. 118-125, 1995

[3] David Borland, Russell M. Taylor, “Rainbow Color Map (Still) Considered Harmful”, Computer Graphics and Applications, IEEE, Volume 27, Issue 2, pp. 14-17, 2007

[4] Cynthia A. Brewer, “Color Use Guidelines for Data Representation”, Proceedings of the Section on Statistical Graphics, American Statistical Association, Alexandria VA. pp. 55-60, 1999

[5] Cynthia A. Brewer, “Designing better maps. A Guide for GIS Users”, ESRI Press, 2005

[6] Cynthia A. Brewer, http://www.colorbrewer.org, 2006

[7] CIE, “Recommendations on Uniform Color Spaces – Color-Difference Equations – Psychometric Color Terms”, Supplement No. 2 to CIE Publication No.15, Central Bureau of the CIE, 1979

[8] CIE, “Colorimetry Second Edition”, CIE Publication No. 15.2, Central Bureau of the CIE, Vienna, 1986.

[9] CIE, Commission Internationale de l’Eclairage, http://www.cie.co.at, 2007

[10] Mark D. Fairchild, “Color appearance models”, Addison-Wesley, 1997

[11] Christopher G. Healey, “Choosing Effective Colors for Data Visualization, Proc. IEEE Visualization, IEEE CS Press, pp. 263-270, 1996.

[12] Ross Ihaka, “Colour for Presentation Graphics.”, Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria, 2003

[13] Haim Levkowitz, Gabor T. Herman, “The Design and Evaluation of Color Scales for Image Data”, IEEE Computer Science and Applications, Volume 12, No. 1, pp. 72-80, 1992

[14] Haim Levkowitz, “Perceptual steps along color scales”, International Journal of Imaging Systems and Technology, Volume 7, No. 2, pp. 97-101, 1996

[15] Bruce Justin Lindbloom, http://www.brucelindbloom.com, 2007

[16] Bruce MacEvoy, http://www.handprint.com/HP/WCL/wcolor.html, 2005

[17] MagnaView, http://www.magnaview.nl, 2007

[18] Binh Pham, “Spline-based Color Sequences for Univariate, Bivariate and Trivariate Mapping”, IEEE Visualization, Proceedings of the 1st conference on Visualization '90, pp. 202-208, 1990

[19] Stephen M. Pizer, John B. Zimmerman, "Color display in ultrasonography", Ultrasound in Medicine and Biology, Volume 9, No. 4, pp. 331-345, 1983

[20] Charles Poynton, "Color FAQ - Frequently Asked Questions Color", http://www.poynton.com/ColorFAQ.html, 2006

[21] Penny Rheingans, "Task-based Color Scale Design", Proc. SPIE-Int Soc. Opt. Eng., Volume 3905, pp. 33-43, 2000

[22] Philip K. Robertson, John F. O'Callaghan, "The generation of Color Sequences for Univariate and Bivariate Mapping", IEEE Computer Graphics and Applications , Volume 6, No. 2, pp. 24-32, 1986

[23] Bernice E. Rogowitz , Lloyd A. Treinish, "How NOT to Lie with Visualization", Computers in Physics, Volume 10, Issue 3, pp. 268-273, 1995

[24] Bernice E. Rogowitz , Lloyd A. Treinish, "Why Should Engineers and Scientists Be Worried About Color?", IBM Thomas J. Watson Research Center

[25] Guarav Sharma, Wencheng Wu, Edul N. Dalal, "The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations", Color Research & Application, Volume 30, Issue 1, pp. 21-30, 2004

[26] Ian Spence, Natasha Kutlesa, David L. Rose, "Using Color to Code Quantity in Spatial Displays", Journal of Experimental Psychology: Applied, Volume 5, No. 4, pp. 393-412, 1999

[27] Maureen C. Stone, "Representing Colors as Three Numbers", Computer Graphics and Applications, IEEE, Volume 25, Issue 4, pp. 78-85, 2005

[28] Bruce E. Trumbo, "Theory for coloring bivariate statistical maps", The American Statistician, Volume 35, Issue 4, pp. 220-226, 1981

[29] Roel Vliegen, Jarke. J. van Wijk, Erik-Jan van der Linden, "Visualizing Business Data with Generalized Treemaps", Transactions on Visualization and Computer Graphics, Volume 12, No. 5, pp. 789-796, 2006

[30] Colin Ware, "Color sequences for Univariate Maps: Theory, Experiments, and Principles", IEEE Computer Graphics and Applications, Volume 8, No. 5, pp. 41-49, 1988

[31] Colin Ware, "Information Visualization, perception for design", Morgan Kaufmann, 2004

[32] Achim Zeileis, Kurt Hornik, "Choosing Color Palettes for Statistical Graphics", Research Report Series / Department of Statistics and Mathematics, No. 41, Wien, Wirtschaftsuniv., 2006

[33] Hongqin Zhang, Ethan D. Montag, "Perceptual Color Scales for Univariate and Bivariate Data Display", Proceedings of the 30th International Congress on Imaging Science: ICIS '06, pp. 115-118, 2006