

The Making of Continuous Colormaps

P. Nardini, M. Chen, *Member, IEEE*, F. Samsel, R. Bujack, M. Böttinger, and G. Scheuermann, *Member, IEEE*

Abstract—Continuous colormaps are integral parts of many visualization techniques, such as heat-maps, surface plots, and flow visualization. Despite that the critiques of rainbow colormaps have been around and well-acknowledged for three decades, rainbow colormaps are still widely used today. One reason behind the resilience of rainbow colormaps is the lack of tools for users to create a continuous colormap that encodes semantics specific to the application concerned. In this paper, we present a web-based software system, CCC-Tool (short for *Charting Continuous Colormaps*) under the URL <https://ccctool.com>, for creating, editing, and analyzing such application-specific colormaps. We introduce the notion of “colormap specification (CMS)” that maintains the essential semantics required for defining a color mapping scheme. We provide users with a set of advanced utilities for constructing CMS’s with various levels of complexity, examining their quality attributes using different plots, and exporting them to external application software. We present two case studies, demonstrating that the CCC-Tool can help domain scientists as well as visualization experts in designing semantically-rich colormaps.

Index Terms—CCC-Tool, charting continuous colormaps, colormap specification, perceptual uniformity, colormap analysis.

1 INTRODUCTION

DESIGNING colormaps is one of the most abiding and prevailing topics in the field of visualization. The critiques of rainbow colormaps by Rogowitz and Treinish [1], Borland and Taylor II [2], and many others have been discussed in almost every course on data visualization, and in the tutorial on color theory by Rhyne [3] that has been a fixture in IEEE VIS for many years. Online software systems for recommending colormaps, such as *ColorBrewer* [4], have been popular among visualization researchers and practitioners. Many practitioners of different application domains prefer discrete colormaps for mapping continuous data by filled contours (e.g., [5]). However, in recent years, there was a noticeable amount of research effort on categorical colormaps (e.g., [6], [7], [8], [9]) as well as a refreshed interest in continuous colormaps (e.g., [10], [11], [12]). As part of this continuing collective effort by the visualization community, this work focuses on the specification of continuous colormaps.

While the critiques of rainbow colormaps are widely acknowledged, they are still widely used. Many domain scientists are aware of more effective choices (e.g., [5]). However, there are many possible reasons why rainbow colormaps are still often employed. Some visualization tools use rainbow as a default colormap. Further explanations are, e.g., the use of legacy code, the need for compatibility to existing visualizations, the effort needed to customize

colormaps, or lacking guidance on effective alternatives. Commonly used visualization software contain as many as one hundred predefined colormaps. These colormaps, however, do not encode domain-specific semantics (e.g., points of freezing, melting, boiling, etc.), various categorizations (e.g., periods, phases, wavelengths, quality levels, etc.), and various critical values (e.g., between safe and unsafe, normal and abnormal, and profit and loss, etc.). To account for the actual data visualized, predefined colormaps may need to be modified and adapted to domain specific semantics. While there are colormap creation tools available, the above characteristics are difficult to attain [11], [13], [14]. Thus users often default to the traditional rainbow, cool warm and viridis colormaps, in which they are accustomed to the association between colors and domain-specific semantics. Other existing software systems (e.g., [11]) use interactive exploration techniques for the creation of colormaps. While this is beneficial for single cases, the resulting colormaps might not be optimal for different datasets captured with similar mechanisms, for automated generation of visualizations in routine activities.

Without a protocol and an easy-to-use tool for specifying and analyzing colormaps, many application users could not propose and deliberate any new standard colormap without writing programs for generating and analyzing new colormaps. In the field of visualization, there is a consensus that a continuous colormap should provide perceptually-uniform mapping from values to colors [10], [12], [15]. However, perceptually-uniform mapping comes with a trade-off with resolving power. The value of the resolution versus the equality can have a very significant impact on the information revealed, and this is a decision for the scientist. Limited software for the creation of perceptually-uniform colormaps or the analysis of colormaps to perceptually-uniformity hinders the creation of such colormaps. Furthermore, implementing this wisdom is curtailed by current limitations in color science and technology. All relatively sophisticated color difference metrics are only considered

- Pascal Nardini is with Institute of Computer Science, University of Leipzig, Germany. E-mail: see nardini@informatik.uni-leipzig.de
- Min Chen is with Department of Engineering Science, University of Oxford, UK. E-mail: min.chen@oerc.ox.ac.uk
- Francesca Samsel is with Center for Agile Technology, University of Texas at Austin, USA. E-mail: fgs@cat.utexas.edu
- Roxana Bujack is with Data Science at Scale Team, Los Alamos National Laboratory, Los Alamos, NM, USA. E-mail: bujack@lanl.gov
- Michael Böttinger is with German Climate Computing Center (DKRZ), Hamburg, Germany. E-mail: boettinger@dkrz.de
- Gerik Scheuermann is with Institute of Computer Science, University of Leipzig, Germany. E-mail: scheuermann@informatik.uni-leipzig.de

Manuscript received Month Day, 2018; revised Month Day, 2018.

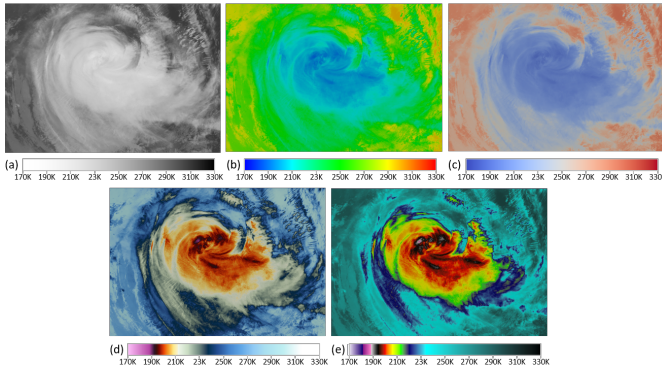


Fig. 1: Given a scalar field, e.g., a temperature field of a hurricane in (a), we all agree a rainbow colormap is not the best solution (b). We can also observe the commonly-used blue-white-red map (c) is unsuitable as the semantics of using white to represent 250 Kelvin can cause confusion. Using more color bands with brightness variation helps (d). As shown in (e), Scientists introduced some semantics into the colormap, such as discontinuous transitions at 192K and 189K as well as a very wide color band between 225-330.

to be correct in the precisely-controlled laboratory testing environments.

In this paper, we propose to separate the specification from the realization of a colormap. The latter is traditionally given in the form of an array of gradually-changing colors or a hard-coded color-mapping function. We propose that the specification should focus primarily on encoding domain-specific semantics, and on enabling application users to address the aforementioned obstacles. We have developed a web-based software tool, called **CCC-Tool** (*Charting Continuous Colormaps*), based on the idea of a general tool for the creation and export of colormaps with the effort to minimize the needed interaction components. This foundation of the tool is extended by a collection of recommended colormaps, which can be adapted to encode the users own domain-specific semantics, and the facilities for analyzing colormaps. CCC-Tool supports a number of commonly-used color spaces and color difference metrics, currently including RGB, HSV, CIELAB, CIE94, CIEDE2000, and DIN99.

CCC-Tool complements the existing web sites (e.g., [16], [17], [18]) and software systems (e.g., [11], [19]) that recommend continuous colormaps by enabling users to adapt the recommended design ideas more easily. CCC-Tool also complements the existing color-based data exploration tools, such as ColorMoves [11]. On the one hand, tools such as ColorMoves allow users to explore various colormaps that can depict interesting patterns. On the other hand, CCC-Tool allows users to specify a standard or de facto standard based on a collection of default colormaps resulting from exploring specific datasets. Here it is helpful to point out that a standard or de facto standard map is designed to enable the visualization of all possible datasets that may be captured with similar mechanisms. So a standard map is not necessarily optimized for a known dataset, but should be optimized in general for all possible datasets in relation to various visualization tasks that may be performed using the colormap.

2 RELATED WORK

Colormapping. A popular method for visualizing a scalar field is *colormapping* [12], [20], [21]. A colormap defines a path through a colorspace [10]. A narrow interpretation of “continuous colormap” implies the continuity of the colormap, while a broad interpretation implies the continuity of the scalar field, while the path can be comprised of one or more path-segments. In this work, we adopt the broad interpretation.

Color Spaces. The space of all colors is in principle infinite dimensional, but since the human eye has only three receptors for color, the space of distinguishable colors is only three-dimensional [22], [23]. Depending on the choice of the three basis dimensions, many different colorspace were developed. The XYZ space by the Commission Internationale de L’Eclairage (CIE) is considered as the basis of all modern color spaces [24], [25]. It embeds all visible colors unambiguously into one space of three imaginary primaries [26], [27]. A number of spaces, e.g., CIELAB, CIELUV, and DIN99, CIECAM [25], [28], [29] were defined as transformations of XYZ in order to derive an ideal color space [30], where the Euclidean distance is proportional to the perceived color difference. The state of the art in color science indicates that such an ideal color space is yet to come because of the complexity of human vision [31], [32], [33]. Some color standards, e.g., CIELAB with ΔE_{1994} and ΔE_{2000} , address this problem by introducing a non-Euclidean metric on top of a color space [34], [35], [36]. It is widely advocated to use a perceptually *near-uniform* color space for the design, assessment, and improvement of colormaps [37], [38], [39], [40], [41], [42].

Design Guidelines. Many guidelines for colormap design have been proposed. Some have been applied intuitively for centuries [43]. The most popular ones include discriminative power, uniformity, order, smoothness, and monotonicity [10].

Discriminative power [37], [39], [44], [45], [46], [47], [48] refers to the length of a colormap’s path through a color space. It may be measured by the just-noticeable-differences along the path, or coarsely the number of distinguishable colors. The bigger the discriminative power, the higher contrast among pixels with different data values and the more details of the data can be observed. This guideline is sometimes in conflict with other needs such as color symbolism, *order*, and *monotonicity*.

Uniformity [1], [2], [40], [41], [49], [50], [51], [52] describes how well perceived color differences resemble differences in the data values. A perceptually near-uniform colormap represents the underlying patterns of the data more truthfully. This guideline is not always followed due to the difficulties mentioned in Section 1 and the early part of this section.

Order [15], [44], [50], [53], [54], [55] commonly refers to the easiness for observers to determine the order of different colors preattentively or based on learning [10]. It can be measured through experiments where observers are presented a subset of colors from a colormap and given the task to sort them in order. A colormap is considered to be more ordered if more observers could sort the colors in the same order as defined in the colormap. This guideline

is usually followed within each individual color band in a colormap, but not always applied to the colors across different bands.

Smoothness [37], [47], [48], [50] refers to a colormap without discrete bands or sudden color changes. It prevents the misinterpretation of discontinuity when the underlying data is continuous. This guideline is sometimes intentionally violated in order to observe the nominal categorization of continuous data or critical values in the data.

Monotonicity assures the changes of colors within a band depend on only one color attribute. Many advocate monotonicity in luminance [1], [2], [38], [51], [54], [56], [57], [58], [59], [60], [61], but some also suggest monotonicity in hue, saturation, or simple paths in the RGB space [39], [46], [62]. The guideline relates to the guideline of order, but is in conflict with *discriminative power*.

A good summary of design guidelines is given by Rheingans et al. [59] and a taxonomy that differentiates the contexts in which these guidelines have been used is discussed by Bujack et al. [10]. As discussed above these design guidelines are not absolute laws. Some of them are competing and a trade-off needs to be found. In some applications, one or more guidelines have to be violated in order to meet domain-specific needs [63], [64].

Rainbow. The rainbow colormap remains to be one of the most used colormaps even though many visualization researchers have described its potential shortcomings for decades [1], [2], [37], [48], [54]. It transverses the color spectrum in the order of wavelength [2], [45], [65]. One problem is that this order is not universally perceived [45], [46], [59], [62], [65]. If presented with just a few of its colors arbitrarily chosen, observers tend to sort them very differently, e.g., by luminance or alphabetically [2], [15]. Another problem is its lack in perceptual uniformity. In some bands (especially around green), colors are transversed much slower than others, hiding features in the data. The inconsistent variation of luminance causes inconsistent banding that distorts the perception of data changes [1], [2], [54].

Tools. Most popular visualization tools like **ParaView** [66] and **VisIt** [67] have a basic integrated colormap editor. But except for having defaults, they do not provide any guidance for the user. The visualization community provides tools for the guided selection of colormaps. Pioneering work in this realm has been done in **PRAVDA Color** [57]. Here, colormaps are suggested based on the visualization task, data types, and spatial frequency. Also **ColorBrewer** [13] provides carefully designed discrete color palettes and recommendations. **I want Hue** uses custom restriction of hue, chroma, and luminance to calculate discrete color palettes based on the CIE Lab colorspace. A different approach is followed by the tools **Color Hex**, **Adobe Color CC**, or **COLRD**. They provide mechanisms so users can generate, share, rate and/or comment palettes. **ColorCAT** [14] extends **PRAVDA Color** and also encompasses for combinations of visualization tasks. There is an extension to matplotlib [68] called **VisCM**. Based on CIECAM, the user can create near-uniform colormaps that increase linearly in luminance, like the new matplotlib default **Viridis**. However, the emphasis on monotony in luminance restricts the users from meeting other requirements in their applications. **ColorMoves** [11]

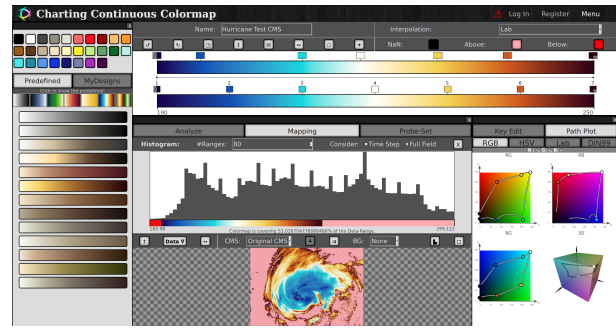


Fig. 2: The screenshot shows a version of Edit-Section of the CCC-Tool. On the left side, the Structure-Window offers constant bands at the top and below a set of predefined CMS or CMS from the MyDesigns list. The CMS-Window at the top right side shows the linear- and below the Band Sketch-visualization of the CMS. At the bottom are the Analysis-Window (left) and Edit-Window (Right), which shows a collection of utilities for Analysis, Color-Map, Mapping, Probe-Sets, Key-Edit and Path-Plot-Edit. Most of these utilities will be detailed in sections 5 and 6.

enables scientists to build colormaps visually customized to align with their data facilitating feature identification and exploration as well as colormaps specifically designed for communication purposes. The custom colormaps are built by assigning pre-designed colormaps to specific regions within the data and seeing the results in real-time on one's data. While this enables the construction of colormaps tuned to a specific data set, it is not designed to produce colormaps for general use. Being a customizing color construction tool it does not contain a means for evaluating the colormaps for widely applicable usage. **ColorMeasures** [10] analyzes the performance of a colormap based on popular design rules evaluated in different color spaces. While this informs the user on characteristics such as discriminatory power, uniformity and the other criteria in Section 2.3, it does not include the ability to create colormaps.

In this paper, we present our web-based colormap tool, by which we can reach a cross-platform ability that only requires an installed browser. This not only improves the accessibility towards tools like Paraview, VisIt, ColorCAT or VisCM but also enables a possible future publication functionality (sharing, rating, commenting) like it is done with discrete color palettes at tools like ColorHex, Adobe Color CC or COLRD. CCC-Tool complements the webtool ColorMoves, which allows to create colormaps exploratively per dataset, with the opportunity to build standardized colormaps for consistent visualizations. In our knowledge, this is the first tool that combines the capability of generating, analyzing, and editing colormaps. This combination empowers the user to make informed decisions about where to closely follow the design rules and where to break them for the benefit of application specific conventions, tasks, audiences, or visualization goals.

3 REQUIREMENTS ANALYSIS AND SYSTEM OVERVIEW

The obstacles discussed in Section 1 provide the motivation for designing and developing a software tool that can enable

easy creation of continuous colormaps for different visualization applications. Our requirement analysis, which was partly based on established expertise in colormaps within the team [6], [10], [11], [52], [64], [69], [70], [71], [72] and partly based on the feedbacks from domain scientists (including one co-author) in different applications, identified the following requirements:

- R1. A colormap can consist of multiple bands to meet the semantic needs of many applications as well as to capitalize confirmed cognitive advantages [56], [69].
- R2. A colormap can feature major transition points that represent important semantic values specific to an application.
- R3. A colormap can have a continuous color transition between two bands in order to facilitate depiction of the gradient of color transformation.
- R4. A colormap can have a discrete color transition between two bands in order to facilitate depiction of contour lines without additional algorithms.
- R5. A colormap can map a value range to a constant color (e.g., for depicting erroneous values).
- R6. Key elements of the colormap have freely selectable successive reference values.
- R7. A colormap can be edited without the need for modifying a large number of interval colors or writing a program to do so.
- R8. Users can browse a library of recommended colormaps, adopt or adapt them for their own applications, and use them as components for designing more complex colormaps.
- R9. Users can choose a color space (from some commonly-used color spaces and some perceptually-uniform color spaces) to interpolate colors across a color band.
- R10. Utilities are provided for users to measure and evaluate attributes of a colormap and visualize these attributes.
- R11. Users can export a conventional colormap (i.e., in an array of value-color intervals) to be used in applications where linear interpolation can be used to interpolate between intervals without the need for complex algorithms associated with some color spaces.
- R12. Ideally the system is available on the web to reach a large number of users around the world.
- R13. The system will be supported by online tutorials on using the system as well as guidance on designing colormaps.

Our web-based system was developed according to the above requirements. Fig. 2 shows a screenshot of the Edit-Section section of the CCC-Tool. We will describe the functions of the CCC-Tool in detail in Sections 4–6. The windows at the screenshot and the other sections of the tool addressing different requirements, such as: Edit-Window and Structure-Window for R1-R9 and Analysis-Window for R10, Gallery-Section for R8, Export-Section for R11 and Tutorial-Section for R13. The section MyDesigns stores up to ten user created colormaps and allows to

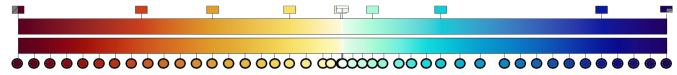


Fig. 3: This figure shows an example of a common colormap (bottom) with a high number of interval colors in comparison to a reproduction (top) according to the CMS idea of less key colors, for creation and editing, and an automatic calculation of many interval colors during the export. (Source: Paraview; Warm to Cool Extended)

manage them. In relation to the obstacles mentioned in the introduction, requirement R8 allows selecting a predefined colormap with specific semantics. R1, R2, R7 facilitate the creation of new colormaps with semantic correlations, or the modification of existing ones. Critical values can be highlighted with R2 or R4. With R6 we ensure the build of standardized colormaps for consistent visualization. The support of perceptually-uniform color spaces allows with R9 and R10 a perceptually-uniform mapping. Furthermore, the idea of the colormap specification allows creating new colormaps, which have the same basic structure and involve the latest future perceptually-uniform color spaces or metrics. CCC-Tool was developed in HTML, CSS and Javascript. It has been tested on the desktop browsers Firefox Version 59.0.1 and Google Chrome Version 64.0.3282.186.

4 COLORMAP SPECIFICATION

Continuous colormaps are traditionally stored and edited as arrays of interval colors (e.g., [19]). Often those colormaps can have a high number of intervals to reach perceptual uniformity, which is very important for software that not able to interpolate in perceptually uniform colorspace (see example at Fig. 3). Another reason to use many interval colors are applications, which use the colormap as a discrete lookup table without linear interpolation. Hence, manually creating and editing such a colormap becomes a time-consuming and tedious task. In most applications where continuous colormaps are deployed, the users are typically interested in controlling the number of color bands, the overall color and their symbolism in each band, the mapping between bands and data ranges, and the transition points (in terms of their colors and data values) between consecutive bands. In almost all situations, users typically wish to observe the gradient of the changes of data values, hence smooth transitions are desirable as shown in Fig. 1(a-d). In many situations, users may wish to be able to relate a pixel color with a particular data range at a glance, hence multiple color bands are desirable as in Fig. 1(b-d). Often one aspect of data semantics (e.g., danger) overwrites the other (e.g., color) as in Fig. 1(d,e). In some situations, users may wish to be able to identify an important data value easily in the visualization, hence a discrete transition between two color bands is useful as in Fig. 1(e). Occasionally, users may wish to map a particular data range (i.e., erroneous data values) to a single color.

All these suggest the necessity for a colormap specification that focuses on bands and major transition points rather than hundreds or thousands of interval colors. Users should be able to incorporate application-specific semantics into the color specification (e.g., the greenish band shows the

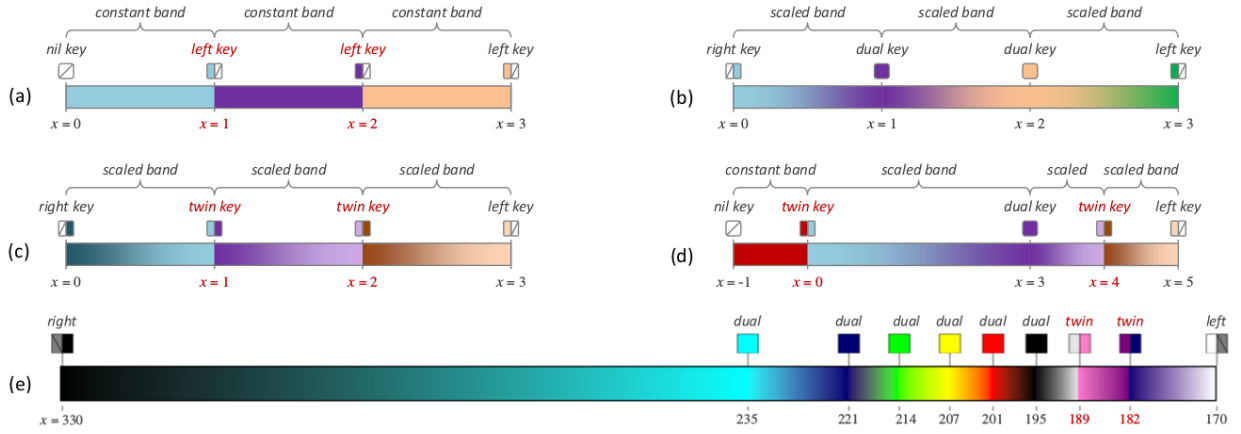


Fig. 4: Examples of creating colormap specifications using different keys. (a) A discrete colormap with 3 discrete colors for 3 data ranges. (b) A continuous colormap with 3 scaled color bands for 3 data ranges. (c) A continuous colormap with two discontinuous transition points (labeled with red texts). (d) A composite colormap features constant as well as scaled color bands and continuous as well as discontinuous transitions. The color bands also have different widths. (e) A CMS that recreates a NASA colormap used for visualizing Hurricane Irma data (7 September 2017).

value range related to a phenomenon X) rather than trying to make some vague associations (e.g., the colors from some shade of light blue to some shade of medium red are related to the phenomenon X).

Hence, in this section, we first consider the novel concept of the *Colormap Specification* (CMS), and then detail its conversion to the traditional interval-based colormap with discussions on color spaces and discrete transitions.

4.1 High-level and Flexible Specification

A colormap specification (CMS) consists of one or more *color bands* B_1, B_2, \dots, B_n ($n \geq 1$). Each band B_i is defined by two transition points, k_{i-1} and k_i , which are referred to as *keys*. Any two consecutive bands, B_i and B_{i+1} , share a common key k_i . Therefore, a CMS is stored as a sequence of keys, $k_0, k_1, k_2, \dots, k_n$.

Each key is a data tuple, $k := (\text{type}, x, c_L, c_M, c_R)$, where x is the corresponding data value in the overall data range to be encoded using colors. Let the overall data range be $[x_{min}, x_{max}]$ where $x_{min}, x_{max} \in \mathbb{R}$ and $x_{min} \leq x_{max}$. The x values in all the keys, $k_0, k_1, k_2, \dots, k_n$ in the CMS must satisfy the conditions $x_0 = x_{min} \leq x_1 \leq x_2 \leq \dots \leq x_n = x_{max}$.

There are five types of keys, each of which is associated with different requirements for the three color specifications, c_L, c_M, c_R , referred to as left, middle, and right colors respectively. For a key, k_i , its left color (c_L) is used to interpret the colors in the color band on the left (B_i), and its right color (c_R) is for the color band on the right (B_{i+1}). We will discuss the middle color (c_M) in Section 4.4. Together with the illustration in Fig. 4, the five types are defined as follows.

- **nil key:** $c_L = c_M = c_R = \text{nil}$. This key (k_i) implies that the color at the corresponding transition point x_i is undefined. It relies on the specification of either k_{i-1} or k_{i+1} . This type of key is only allowed to be defined at $x_0 = x_{min}$.
- **right key:** $c_L = \text{nil}$ is undefined while $c_R = c_M$ is defined. This key is also only allowed at $x_0 = x_{min}$.
- **left key:** c_L is defined but $c_R = \text{nil}$ is undefined. c_M must be assigned to c_L or the nil state of c_R . This key

cannot be used at $x_0 = x_{min}$, but can be used for any other key $k_i, i = 1, 2, \dots, n$.

- **dual key:** $c_L = c_M = c_R$ are defined as the same color. This key cannot be used at $x_0 = x_{min}$ or $x_n = x_{max}$.
- **twin key:** c_L and c_R are defined separately, and can have different colors. c_M must be assigned to one of the two colors, i.e., either $c_M = c_L$ or $c_M = c_R$. This key cannot be used at $x_0 = x_{min}$ or $x_n = x_{max}$.

The asymmetry of the left and right keys is because of the need to avoid the situation where a color band is defined by two nil colors, i.e., when a nil color of a left key k_i is followed by that of a right key k_{i+1} . Using a combination of these five types of keys, we can define a CMS with following types of color bands and transition points:

- **Constant Band:** All colors within a band are of the same color. Given a band B_i with two keys k_{i-1} and k_i , the band (B_i) is a constant band if (i) $c_{R,i-1} = c_{L,i}$; (ii) $c_{R,i-1} \neq \text{nil}, c_{L,i} = \text{nil}$; (iii) $c_{R,i-1} = \text{nil}, c_{L,i} \neq \text{nil}$.
- **Scaled Band:** The colors within the band are interpolated between two different colors $c_{R,i-1}$ and $c_{L,i}$. A band (B_i) is a scaled band if and only if $c_{R,i-1} \neq \text{nil}, c_{L,i} \neq \text{nil}$, and $c_{R,i-1} \neq c_{L,i}$.
- **Continuous Transition:** The color at an intermediate key ($k_i, i \neq 0, i \neq n$) is smoothly interpolated in both left and right directions. A key (k_i) is a continuous transition point if (i) k_i is a dual key; (ii) k_i is a twin key with the same left and right colors, i.e., $c_{L,i} = c_{R,i}$ and (iii) k_i is a left key and its left color ($c_{L,i}$) is the same as the right color ($c_{R,i-1}$) of the preceding key k_{i-1} .
- **Discontinuous Transition:** The color at an intermediate key ($k_i, i \neq 0, i \neq n$) is discontinuous in either left or right direction. A key (k_i) is discrete transition if (i) k_i is a twin key, $c_{L,i} \neq c_{R,i}$; and (ii) k_i is a left key and its left color ($c_{L,i}$) differs from the right color ($c_{R,i+1}$) of the succeeding key k_{i+1} .

It is necessary to note that the widths of color bands do

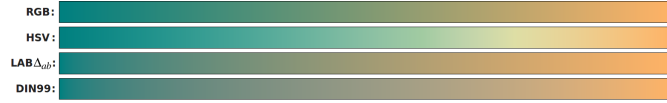


Fig. 5: The image shows the interpolation between two colors in different color spaces. We can observe their visual differences.

not need to be the same. They can be stretched by setting the values of x_i at a non-uniform pace. As shown in Fig. 4, the flexibility this scheme allows users to define a CMS composed of different types of color bands and different transitions. Before 2017, NASA used a multi-band colormap with continuous transitions for showing temperatures of hurricanes. When Hurricane Irma struck Caribbean islands, NASA introduced two new color bands with discrete transition points for the range [170-189] Kelvin (Fig. 4(e)). Our scheme can easily address such needs.

4.2 Color Spaces

For any CMS with scaled color bands, the corresponding colormap may appear differently depending on the color space used for interpolation. Fig. 5 shows four different interpolations between a cyan color ($\text{RGB}_{[0,1]} : 0, .5, .5$) and an orange color ($\text{RGB}_{[0,1]} : 1, .7, .4$) using the RGB space, HSV space, CIELAB with CIEDE2000, and DIN99.

The CCC-Tool allows to define each color using one of the four spaces, namely RGB, HSV, CIELAB, and DIN99. Given two colors in RGB space (or HSV), the CCC-Tool uses linear interpolation to define the color transformation within the color band. For the CIELAB space, there are several ways to measure the color difference between two colors, including Euclidean distance (ΔE_{ab}), CIE94 (ΔE_{94}), and CIEDE2000 (ΔE_{00}), representing the progressive improvement of the perceptual uniformity of the difference metrics. The CCC-Tool supports all three metrics.

DIN99 relates closely to CIELAB and it provides a transformation from (L, a, b) to (L_{99}, a_{99}, b_{99}) so that the color differences can be computed in a Euclidean distance metric. The difference measures using DIN99 are perceptually more uniform than those using CIE94 and are comparable to those using CIEDE2000 [73], [74]. Because of the transformation, given two colors in the DIN99 space, the CCC-Tool also uses linear interpolation. In this work, we leave the decision about the color space to users, while we provide users with utilities to analyze and compare CMSs using different color spaces.

4.3 CMS to Colormap Conversion

As we mentioned before, a colormap is traditionally stored as an array of interval points, $[z_\tau, c_\tau]$, $\tau = 0, 1, \dots, \eta$, where z_τ is a reference value in the data range, c_τ is a color, and $\eta + 1$ is the number of intervals. In general, most application software uses such a colormap as a look-up table. Given a value v in the data, the software first searches for two consecutive cells in the array such that $z_\tau \leq v \leq z_{\tau+1}$. The software may determine the color of v by selecting the nearest cell, in other words, c_τ if $|z_\tau - v| \leq |v - z_{\tau+1}|$, and $c_{\tau+1}$ otherwise. Alternatively, the software may interpolate the color linearly between the two cells based on $z_\tau, v, z_{\tau+1}$.

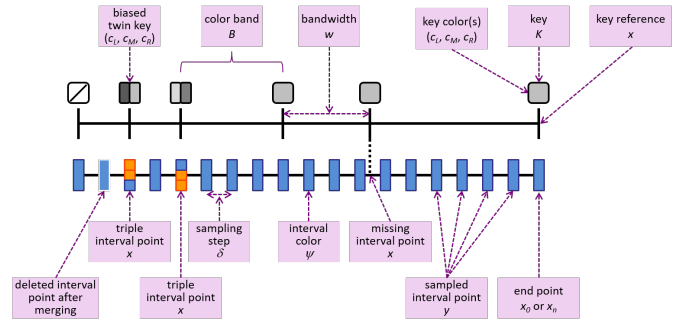


Fig. 6: The merge of the reference points in a CMS and a set of regularly sampled intervals to create a final set of intervals for a colormap.

Most applications do not directly support less common color spaces such as CIELAB or DIN99, or non-Euclidean difference metrics. In order to support such applications, CCC-Tool allows exporting a CMS to a conventional colormap in the form of an array of sampled colors. Fig. ?? shows a screenshot of the export page where a constructed CMS is converted to a conventional colormap. The user can choose a color space for defining colors from options RGB, HSV, LAB (CIELAB), and DIN99. The user can also select a color space for interpolation from RGB, HSV, LAB (i.e., CIELAB with ΔE_{ab}), de94 (i.e., CIELAB with CIE94), de2000 (i.e., CIELAB with CIEDE2000), and DIN99. Colormaps can be exported as XML, JASON, or CSV.

One may uniformly sample the data range $[x_{min}, x_{max}]$ to determine the reference values z_τ in each cell of the colormap. For example, one may start with $y_0 = x_{min}$, and add more interval points with a uniform step δ as $y_{j+1} = y_j + \delta$, until reaching y_m such that $y_m + \delta > x_{max}$. However, this is not adequate enough. As illustrated in Fig. 6, some reference values of the keys in the CMS, $x_i, i = 0, 1, 2, \dots, x_n$ may not coincide with the sample points. We thus need to merge the two sets of reference values $\{x_0, x_1, \dots, x_n\}$ and $\{y_0, y_1, \dots, y_m\}$ into the z_0, z_1, \dots, z_η . The merge also ensures that $z_\eta = x_{max}$.

4.4 The Function of Middle Color c_M in Keys

Whenever applying a colormap with discontinuous transitions to a continuous data range, there is a common issue of consistency. As each discontinuous transition point x is associated with two different colors, their interpolation is inevitably ambiguous. In traditional colormaps, this issue is usually addressed by having two overlapping sample colors. For example, consider the following four color cells, $\mathbf{A}, \mathbf{B}_L, \mathbf{B}_R$, and \mathbf{C} , in a colormap: $\dots, \mathbf{A} : [z_{j-1}, c_{j-1}]$, $\mathbf{B}_L : [z_j, c_j]$, $\mathbf{B}_R : [z_{j+1}, c_{j+1}]$, $\mathbf{C} : [z_{j+2}, c_{j+2}]$, \dots , where \mathbf{B}_L and \mathbf{B}_R are at the transition point x such that $z_j = z_{j+1} = x$ and $c_j \neq c_{j+1}$ at the transition point.

This approach makes the ambiguity largely unnoticeable. In most application software, mapping the value x involves a search for two interval colors. If the software searches from $j = 0$, it will reach the transition point from the left by finding \mathbf{A} and \mathbf{B}_L . So the color for x will be c_j . If a user unknowingly interprets c_{j+1} in \mathbf{B}_R as x exactly, what is depicted by c_{j+1} is actually associated with $x + \epsilon$ where $\epsilon > 0$ is a small number. Occasionally, especially with a

discrete colormap, the problem can be serious as illustrated in Fig. 7.

Fig. 7 shows a simple colormap corresponding to three discrete bands. The colormap is represented by six interval points, $\mathbf{A}, \mathbf{B}_L, \mathbf{B}_R, \mathbf{C}_L, \mathbf{C}_R, \mathbf{D}$, where \mathbf{B}_L and \mathbf{B}_R are with the transition point at $x = 2$, and \mathbf{C}_L and \mathbf{C}_R are with $x = 4$. Given the example dataset, if we search the colormap from the left, we get a cyan-white visualization, and if we search from the right, we get a white-orange visualization. The user may want the first transition point to be cyan and the second to be orange in order to have a cyan-white-orange visualization.

Although this problem is inherently that of conventional colormaps, we have devised a solution to the problem through our CMS scheme. We introduce the middle color c_M in twin keys and left keys. This field can be used to specify the exact color of the transition point. c_M is not really a new color, but a toggle between c_L or c_R .

When such a CMS is exported to a colormap, for each transition point we export three interval colors (instead of two) as: $\dots, \mathbf{B}_L : [z_j, c_j], \mathbf{B}_M : [z_{j+1}, c_{j+1}], \mathbf{B}_R : [z_{j+2}, c_{j+2}], \dots$, where $z_j = z_{j+1} = z_{j+2} = x$ is the transition point, and c_j is the left, c_{j+1} the middle, and c_{j+2} the right color. Through its help and tutorial facilities, the CCC-Tool provides users with a mapping algorithm that will check for the middle color c_M , while working as usual for any other conventional colormaps. Given an input value v and a colormap $[z_j, c_j], j = 0, 1, 2, \dots, \eta$, the algorithm returns a color for v .

Algorithm : Color-Mapping (Middle of Triple)

```

1: check: within the data range
2: if  $v > z_\eta$  or  $v < z_0$  then
3:   return error
4: loop
5: for  $i = 1$  to  $i = \eta$  step 1 do
6:   if  $v = z_i$  then
7:     if  $i < \eta$  and  $v < z_{i+1}$  then
8:       if  $i < \eta - 1$  and  $v = z_{i+2}$  then // transition point
9:         return  $c_{i+1}$  // middle color
10:      else // conventional colormap
11:        return  $c_i$ 
12:      else // conventional colormap
13:        return  $c_i$ 
14:    $c := (c_i - c_{i-1}) * (v - z_{i-1}) / (z_i - z_{i-1}) + c_{i-1}$ 
15:   return  $c$ 

```

4.5 Probes

When a small sub-range of the global data range needs to be inspected or highlighted in more detail, *probes* can be defined within the tool and combined with any CMS. In our context, a probe is a sub-range within a continuous colormap that is replaced by or blended with either a uniform color (Constant-Probe), a color gradient (One-Sided-Probe), or a color/transparency gradient (One-Sided-Transparent-Probe). In resulting visualizations, probes (or sets of probes) have the appearance of isoline-like structures of variable width. The width visualizes the spatial data gradient - the larger the width, the smaller the

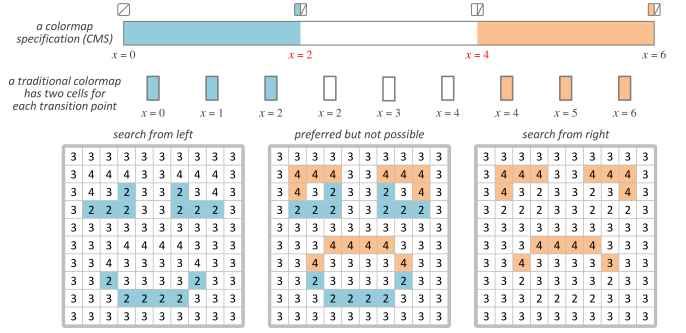


Fig. 7: Illustration of the consistency problem in colormap mapping. Given a traditional colormap with two discontinuous transition points at $x = 2, 4$. When a conventional colormap algorithm searches the color table from left, it generates the image on the left. When it searches from right, it generates the image on the right. It would not be possible to generate the middle image. The middle color c_M in our CMS addresses this issue.

gradient is. Using a gradient shading for probes simplifies the depiction of the local gradient within the global data range.

Within CCC-Tool, a probe is defined in addition to the normal colormap part of the CMS. This allows displaying the data with the "normal" CMS and to overlay the probe on request. However, since the colormap definitions of visualization systems so far don't support directly using a CMS that include separate probe definitions, two colormaps need to be exported, the original CMS and the original CMS combine with the probe(s). Both can then be imported and used within visualization tools such as ParaView. For interactive probing, the probe feature and means to change the probe settings interactively would have to be implemented directly in the visualization systems used. Since the computation of isolines additional to colormapping increases the computational costs, the use of probes via colormapping could be advantageous especially for interactive inspection of very large data sets.

All probes are defined by a starting value \mathbf{S} , an ending value \mathbf{E} and a probe-type. For the probe-types One-Sided and One-Sided-Transparent, the definition also includes the function $f_p(\mathbf{x}, \mathbf{c})$ with x is a data value within the probe. The function defines a process (0 to 100 or 100 to 0) of saturation or lightness by linear interpolation between \mathbf{S} and \mathbf{E} . Furthermore the probe definition of Constant and One-Sided-Probes includes a color \mathbf{C}_p . Adding a Constant-Probe to a CMS requires the update of keys or an extension of new ones to get a left-key at \mathbf{S} and twin-key with $c_L = C_p$ at \mathbf{E} . The creation of a One-Sided-Probe needs a twin-key with $c_R = f_p(\mathbf{S}, C_p)$ at \mathbf{S} and a twin-key with $c_L = f_p(\mathbf{S}, C_E)$ at \mathbf{E} . One-Sided-Transparent-Probes are determined according to the same method of the One-Sided-Probe, except that the algorithm uses the CMS-colors C_S at \mathbf{S} and C_E at \mathbf{E} instead of C_p . The colors of CMS-keys, located between \mathbf{S} and \mathbf{E} , have to be updated with $f_p(x, c)$. With the monotone decreasing or increasing of lightness or saturation, the two One-Sided probe types form an ideal basis to inspect the range of the probe. It is easy to include such probes through the twin key of the

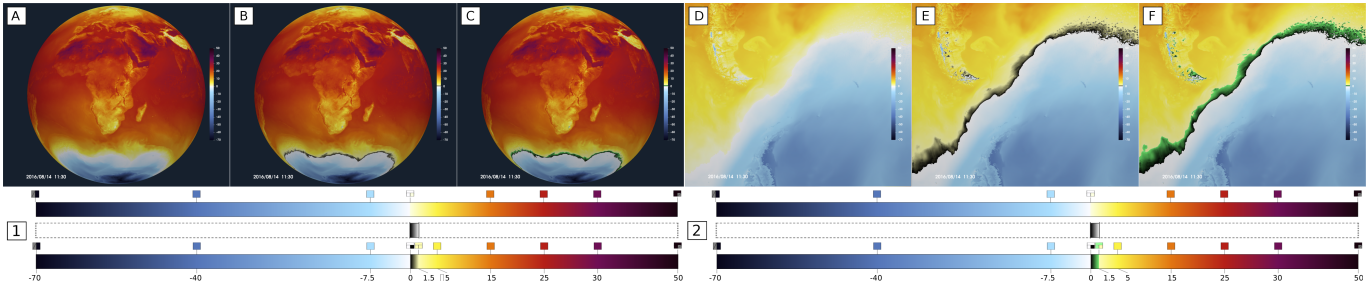


Fig. 8: At the example of the 2m-temperature of a high resolution simulation with the global atmosphere model ICON, the figure illustrates the use of probes to inspect small sub-ranges of the global data range. The rendering on the left (A) shows the the global temperature distribution with color-mapping using a CMS (inset 1, top) that was designed to resolve the data range from -70 to 50°C. However, within small sub-regions, as shown in a close-up (D), only a small section of the CMS is used and local structures are hardly visible. In order to probe the temperature range 0 - 1.5°C in more detail, we added single probe at 0°C to compose a CMS (1, bottom) that creates an isoline-like-structure to highlight the freezing point and the data range above. The images (B) and (E) show the result for a One-Sided-Transparent-Probe. The according colormap composition is shown in inset (1). Similarly, (C) and (F) show the according renderings for One-Sided-Probe according to the definition shown in inset (2, middle). (1: Top: Divergent CMS for the 2m-temperature. Middle: one sided transparent probe for the range 0 - 1.5°C. Bottom: resulting colormap. 2: One sided probe without transparency.)

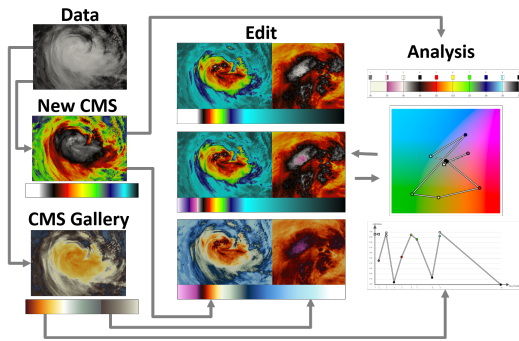


Fig. 9: A typical workflow for creating a colormap specification (CMS) using the CCC-Tool. The example dataset illustrates steps may be used to recreate the colormap used by NASA to visualize hurricane data (courtesy of William Straka III). A user may use the Gallery to choose from a built-in colormap, or the New to create more complex or application-specific CMSs.

CMS with a hard discontinuous transition of lightness to create the impression of an isoline-like-structure (see Fig. 8).

5 CREATING AND EDITING AN CMS

The introduction of a colormap specification (CMS) allows users to create and edit colormaps, and do retrospective amendments without the need for working with hundreds of elements in an array like at conventional programs for generating colormaps. The following paragraphs explain the methods for creation and editing of CMS used by CCC-Tool.

5.1 Creating and Editing a CMS

Fig. 9 shows the typical workflow of using the CCC-Tool. We offer a collection of CMS and a user can use them as initial CMS or directly for the export. Most of these were imported from ColorMoves [11], which were designed to provide discriminatory power within specific hue ranges and hue

combinations. Alternatively, a user can start with an empty CMS. The following create/edit methods are independent, whichever way of beginning the user chooses.

Many users will likely use the CCC-Tool to create a CMS in order to meet the requirements of specific applications. Here we use the colormap created by NASA for displaying hurricane datasets as an example. For a fast and easy compilation of a CMS, we implement sets of predefined CMS for the main edit/creation mechanism of our tool (see Fig. 2). The elements of these sets are drag-able and can be dropped between the key defined bands of the user’s CMS. To ensure the condition of a good overview over the whole CMS during the creation/edit process, we decided to display the CMS on one hand as a corresponding linear plot and on the other hand as Band Sketch, where all bands have equal lengths. In many cases, the user may wish to introduce new bands or remove existing bands. Like at our example, the hurricane Irma (August/September 2017) exhibited unexpected strength. The very low temperature area of the data cannot be observed easily. As reported in the news at that time, scientists combined two new bands together with two discontinuous transition points as shown in the center-middle CMS in Fig. 9. Although the data used here is not that of Irma, we can still see the pink patch with temperature below 189 Kelvin, which is more difficult to see this using the CMS at the top. The discontinuous transition point between white and pink ensures that any data values just below 189 Kelvin will “pop out”, rather than smoothly changing from white to pink. With the drag and drop method of predefined CMS from the Structure-Window we created a process, where the user can easily handle such a situation.

Next to this technique, we also implement Key Edit and Path Plot offered at the Edit-Window, where the user can modify each key, including all its attributes, i.e., $k := (type, x, c_L, c_M, c_R)$. While the first method is a perfect beginning to create a new CMS, the modifying of keys is only practicable for a non-empty CMS. Modifying the *type* of a key allows the user to change the continuity of the corresponding transition point and the appearance of the

two neighboring color bands. Modifying the reference value x allows the user to change the width of the neighboring bands. Modifying the colors c_L, c_M, c_R allows the user to refine detailed color specification to encode application-specific semantics such as color symbolism. In Fig. 9, the top CMS in the middle column shows more hurricane-related semantics being introduced, including different widths of the bands and the modified key colors. As the lower temperature values typically correspond to stronger part of the hurricane, narrower and more-attentive colors were used at the low-temperature side of the CMS.

Back to the hurricane example, we created an alternative CMS as shown at the bottom-middle of Fig. 9. In general, the CMS captures most of many features of the NASA colormap, but the missing of discontinuous transitions makes the observation of critical values hard. With the drag and drop technique, the solution to this problem would be a replacement of the band with a new one. The `Key Edit` offers the more intuitive solution of adding a discontinuous transition point by changing the key type.

In addition, to edit the CMS, we implement an interactive variant of the path plot (see Section 6.2), we called `Path Plot`, to modify the colors of the keys. Such a plot allows the user to manipulate the CMS with respect to a chosen color space. For example, a user may start with a plot in the HSV space as it aligns well with humans' intuitive descriptions about colors. A CMS is shown as a path in HSV and the user can adjust a node on the path by moving it to a location of the desired hue, saturation, and value. Once the user has determined the nodes (i.e., keys) on the path, the user can use the CIELAB space to create a perceptually more uniform CMS.

5.2 General Methodology for Constructing a CMS

Constructing a CMS requires the user to make a series of decisions: (a) how many color bands are needed within the CMS? (b) Are any bands or transition points of specific importance? (c) Does a domain-specific convention, such as color symbolism, exist? Some colormaps are created to assist in solving a specific problem in a specific dataset while others are steps towards a widely used CMS. Below we outline some general guidelines for using the CCC-Tool.

Key and Band Colors. Hue selection can be used to direct attention, define high-level categories of data values, and assign domain-specific semantics (e.g., red for risk) [15], [75]. In order to accomplish this, it is important to budget saturated hues and types of contrast. Allocating saturated and de-saturated hues as well as types of contrast enables the creation of organized visual hierarchy while minimizing distracting contrast [76]. Think of hues as having auditory volume. Fully saturated red is very loud, burnt orange is a mid-volume and tan is low volume. Assign the loudest colors to the most important data range and the quietest to ranges important only for context. If the data values are of equal importance, use hues of equal volume. In Fig. 9, the `Gallery` example places all of the saturation in the area of the hurricane. The "Create New" example renders the hurricane in grays and areas outside the hurricane in saturated lime green thus misdirecting attention. Related data ranges assigned colors close on the color wheel such

as blue, green, and yellow will be intuitively associated. Unrelated ranges rendered complimentary colors such as red and green signal multiple categories [15].

Number, Type, and Widths of Bands. The number of bands typically depends on the domain-specific semantics associated with the data (e.g., the number of categories). When such semantics are not available, more bands correspond to an increase in discriminatory power. However, too many bands can cause the visualization to become confusing. Constant bands are typically used for easy and quick identification of specific values within the data, while scaled bands enable the observation of more detailed patterns of variations in the data. The multiple bands with consistent widths and monotonic change of brightness within each band allow for more accurate observation of the gradient of the changes in the data. Varying widths allows users to observe important range of the data with more narrow bands, while "spending" less colors on less important data ranges.

Domain Convention. Encoding domain-specific semantics enables easy understanding and remembering of a colormap, and facilitates rapid observation and effective communication. Color symbolism is a powerful mechanism (e.g., red for hot or risk). The uses of discontinuous transitions need to be carefully managed. Colormap legends are essential for minimizing misinterpretation.

6 VISUAL ANALYSIS OF CMSs

The current version of the CCC-Tool offers six different visualization methods to analyze a colormap. These analysis plots provide information about interpolation, conditions between color spaces, uniformity, and order. Except for the key speed plot, all analysis plots use information of the interval colors. Fig. 11 shows an example using the speed (6.3) and the order (6.4) to improve a colormap.

6.1 Interpolation Space

For any CMS with scaled color bands, the corresponding colormap may appear differently depending on the color space used for interpolation. The `Interpolation Space` plot informs the user about possible visual differences. The example of Fig. 5 displays four interpolations between a cyan color and an orange color using linear interpolation in the colorspaces RGB, HSV, CIELAB, and DIN99.

6.2 Charting the Path of a Colormap

The `Path Plot` contains information about the key color positions and interval color positions. Likewise, the colormap path in one color space is indicated by an interpolation in another color space. The plot uses combinations of 2D and 3D visualizations. For the HSV-, LAB- and DIN99-path-plots variant, we implemented a plot showing the top view of the colormap path and the colorspace. Furthermore, we produced three plots: one for each component of a color. Their main task is to visualize the path of the colors for each of the components separately. In case of the RGB path plot, all color components include hue and saturation. Therefore, we use a combination of three 2D plots projected along the

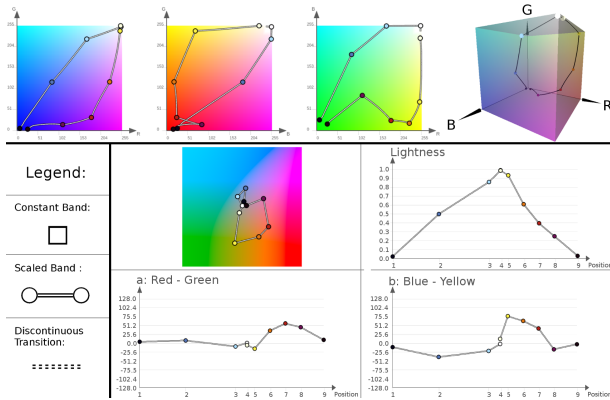


Fig. 10: This image shows two out of four variants of the colormap path plot. The upper one shows the RGB path plot and the bottom example the Lab path plot. Both plots show the CMS from Fig.8 using the Lab Colorspace for interpolation.

axes, and a 3D rendering of colormap path and the space. Fig. 10 shows an example for a colormap path interpolated and visualized in RGB and a second example interpolated in LAB space and visualized in the HSV space. Every scaled band of a CMS is represented in the colormap path by a solid curve and one circle at each line end. The positions and the coloring of the circles are defined by the key colors which delineate the band. The solid lines are constructed with straight lines connecting each color position with the next one including the interval colors which are located between the two key colors. These solid lines form a straight line, if the interpolation space and the analyzed space are the same. If this is not the case, the solid lines result in a curve. Constant bands are illustrated as a single colored rectangle in the top view and RGB plots, and with two rectangles connected with a horizontal solid line in the line plots. Discontinuous transition points are indicated by dashed lines. For the identification of discontinuous transition points, the algorithm has to find twin-keys or left keys with $x_i \neq x_{max}$.

With the path plot, users can envision how the colormap passes through a color space and can identify critical parts like closely located colormap segments or intersections. The management of color components is also helpful for the evaluation like lightness, saturation and hue. The second example of Fig. 10 shows, how we used the plot to create a divergent CMS with a nearly monotonously increasing of the lightness till the key at reference value 0°C and a following decreasing of the lightness.

6.3 Speed Plots

For the analysis of perceptual uniformity, we implement the global and local speed plots according to Bujack et al. [10]. For colormaps with discontinuous points, we use the color of the maximum value for the global speed plot and omit to draw a bar at the local speed plot. Massive speed difference within the colormap leads to indistinguishable coloring of the remaining value ranges. For a better visualization in such cases, we implement a logarithmic mapping next to the linear color mapping. Alternatively, it is possible to focus these plots on a user specified key range. We use alternating dark and light squares for the background to prevent any

region to fade into a single background color. It can be difficult to visually analyze the speed between keys in the global plot. Therefore we also offer additional key speed plots. These figures show a stacked bar graph for the speed from one key to the next one.

6.4 Visualizing the Order of Colors

To analyze the order of colors, we build on the work of Bujack et al. [10], [77]. They evaluate intrinsic order depending on the mutual perceived distances between the colors. We merge the two matrix plots that they use into one newly designed plot (Fig. 11). We draw the negative local order values, which violate order, in red and the positive ones, which follow order, in blue local order values above the colormap. The visualization of the global order shows only the interesting negative values. For each interval of colors c_i and c_j , we calculate the minimal order value of the colors c_k with $i < k < j$. In case of a negative value, we draw an arc from the position of c_i to c_j colored in c_k . The maximal distance from the arc to the colormap shows how severely the intrinsic order is violated.

7 APPLICATION CASES OF CCC-TOOL

7.1 Color Mapping of 3D Ocean Temperature Data

In climate sciences, 3d data representing the Earth system's compartments such as the atmosphere or the ocean are regularly analyzed. Due to different climatic conditions in different geographic regions, the data can, depending on the quantity chosen, span a wide range of values. We use, as shown in Fig. 12, the temperature data of a regional ocean model simulation covering the South Atlantic west of Africa as described in [78]. The Benguela Upwelling System, one of the major coastal upwelling systems [79], [80], transports cold (and nutrient rich) water to the surface near the coast of Namibia.

Our goal here is to design a color palette and a mapping between color and temperature that efficiently emphasizes the relevant structures in the sea surface temperature (SST), but that, at the same time, reasonably resolves the temperature variations found at greater depths. For appropriate visualization of these features with a combination of vertical and horizontal slices (2D) using one joint colormap, we define the following requirements:

- 1) Global data range to be represented: $(0-30^\circ\text{C})$
- 2) The sector of the color map representing the sea surface temperature $(13-30^\circ\text{C})$ requires as many discriminable colors as possible to visualize the complex horizontal flow patterns also represented by SST variations.
- 3) Slight temperature variations in the Benguela Upwelling System (the dark green area east of Namibia shown in Fig. 12A, cf. [78]) need to be resolved in particular.
- 4) The remaining sector of the color map $(0-13^\circ\text{C})$ mainly represents the range of temperatures found at greater depths.
- 5) The colors used in the color map should be linearly ordered and follow best practices used by domain

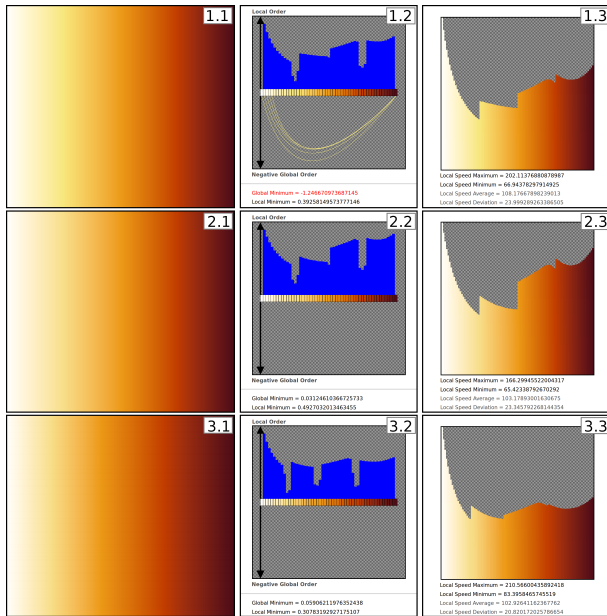


Fig. 11: Illustration of the process of improving an existing colormap. As an example, we chose a variation of the heated body colormap. It is the third from the top in the category “Yellow Orange” in the Gallery of the CCC. The improved colormap suffices intrinsic order w.r.t. ΔE_{2000} , it has a higher minimal discriminative power, and appears smoother. (1.1 Original colormap; 1.2 Order plot of original colormap shows that the CIELAB ΔE_{2000} observer would not be able to order yellow correctly between the bright part and the brown; 1.3 Speed plot of original colormap shows a drop in speed around yellow down to 67; 2.1 Colormap changed to make the yellow less brilliant; 2.2 Order plot shows that the changed colormap suffices global order everywhere; 2.3 Speed plot of changed colormap still shows a drop in speed around yellow down to 65; 3.1 Colormap changed to increase minimal speed; 3.2 Order is not affected by rearranging the locations of the keys; 3.3 Speed plot shows that the minimal speed was raised to 84, which increases uniformity.)

scientists for representing temperature by color, i.e. warm=red, cold=blue.

To get an overview on the data first, we combine several color mapped slices through the 3D model domain using different standard colormaps (Fig. 12 B,C,D) and analyze the features that we observe. First, we look at the horizontal SST patterns at a given point in time.

Within the domain, we find a strong northward increase from about $13^{\circ}C$ in the South-East to $29^{\circ}C$ in the North, overlaid by some local variations caused by ocean eddies and currents. Looking at the vertical East-West-section in the foreground, we find horizontal banding in the temperature with a very strong temperature gradient towards the sea surface. The temperature ranges from little above $0^{\circ}C$ in the deep ocean to about $17^{\circ}C$ at the surface.

With respect to the design of a new colormap optimized for the actual application case, we first visually evaluate the performance of the three standard colormaps used on the right side of Fig. 12 (B,C,D):

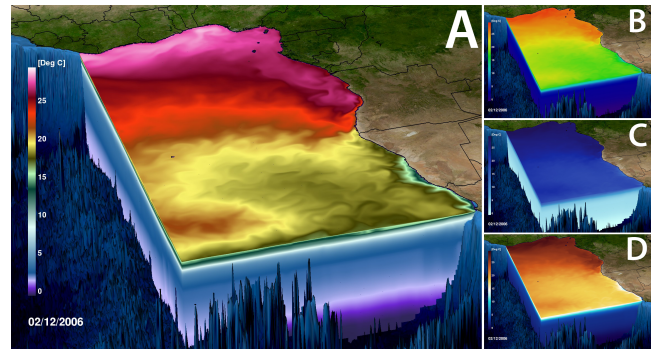


Fig. 12: The visualization of the temperature field of vertical and horizontal slices of a 3d regional ocean model is a good example of the use of multiple color/brightness bands with strong local color and luminance gradients. Note the exaggeration in the vertical scaling to show details in the vertical cross section (A). Compare the complex application colormap with common colormaps: rainbow (B), lightblue to darkblue (C) and blue-white-red (D).

B The rainbow colormap resolves many of the SST patterns described above, but the vertical sections show only few variations except for the strong gradient near the surface. The shades of blue representing low temperatures are hardly distinguishable from the blue color used for rendering the bathymetry. Also, some of the SST structures are not sufficiently represented by color gradients due to lacking variations in luminance. The upwelling zone west of Namibia is hard to depict.

C The strong vertical temperature gradient near the surface is the dominant information in the visualization using the simple scaled band colormap (light blue to dark blue). However, it does not resolve any details in the SST; particularly the cold upwelling zone west of Namibia is not visible.

D The symmetric blue-white-red-colormap shows only little of the temperature variations at the sea surface. Also, only little detail is visible at greater depths. The bright center part of the colormap, which is “loud” due to its high luminance contrast, is only visible on the vertical slices denoting the high vertical temperature gradient.

A For the application case we decided to design a colormap using almost the complete possible hue range combined with luminance modulations. The resulting visualization shown in Fig. 12A illustrates the concept: the luminance waves enhance the local contrast between adjacent temperature subranges. This allows for a detailed visualization of the small scale variations that are related to features in the flow field, and, at the same time, to adequately resolve the full temperature range found in the data. Ranges with strong gradients in the data are represented by quicker changes in the color. The temperature range between 4 and $14^{\circ}C$ is coarser resolved because of the strong temperature gradient found vertically. As a result, the visualization allows depicting all relevant details found in the data.

7.2 Ancient Water Formation

Half of all of the water in the universe predates the sun. Scientists Joseph Smidt, LANL, and Brandon Wiggins, SUU,

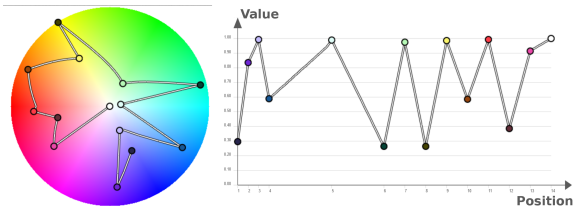


Fig. 13: The Colormap Path analysis of the CCC-Tool applied to the ocean temperature color map visualizes the color and luminance variations used in Fig. 12A.

set out to understand how, where, when and with what frequency this ancient water appears in the early universe. They combined expertise in theoretical astrophysics, molecular chemistry, and high performance computing. Identifying the locations and prevalence of the complex interactions between multiple variables that enable the formation of water is essential as well as closely examining how those variables interact [81]. The team combined a high resolution hydrodynamics simulation of the early universe closely coupled with this hydroxyl and water-producing chemistry model to simulate how water molecules were created and distributed in space and time in the early universe [81].

Answering the scientific questions required tracking, over time, the conditions that give rise to water - dense gas, warm gas temperatures, as well as the presence of oxygen and heavy metals. By examining the densities and distributions of these materials in supernovae explosions, comparative visualization enabled the identification of regions in the ancient universe where water likely formed.

For the simulation, three codes were combined, Enzo, Grackle, and WaterNet, and the results visualized in Paraview. While previous colormaps did little to facilitate the ability to see the relationships between key variables, the CCC-Tool isolated the data ranges within each variable conducive to water formation. Thus the scientists were able to more easily compare multiple variables over time and identify where gas density, temperature, oxygen and heavy metals were co-located within the ranges in which water formation occurs. Specifically, CCC-Tool's ability to apply specific color bands to narrow data ranges within each variable enabled scientists to examine the ensemble visualizations and identifying the locations where multiple variables over time across multiple 2D plots align for potential water formation. Figure 15 illustrates the clarity provided by highlighting the specific heavy metals "sweet spot" for water formation. While the viridis colormap, left, provides an overview of the metals distribution, the cctool colormap clearly identifies the regions scientists need to identify in order to answer their scientific question.

As Figure 14 demonstrates, we used the CCC-Tool to assign a different color palette to each variable, enabling us to focus on different regions of interest. We were able to quickly compare visually the qualities of each variable when conditions aligned to form water.

Focused visualizations proved to be critical here because they not only illuminated critical co-located conditions, but also enable crucial verification and validation throughout the lengthy and complicated coding process. Specifically, the visualizations provided the means for the results to be ver-

ified throughout the process of writing and testing code. In adding the molecular chemistry to Enzo, it is essential to ensure that the chemistry is integrating properly into the code. By comparing visualizations with and without the molecular chemistry, scientists could observe how the chemistry was affecting the behavior of the code. CCCtool enabled scientists to compare simulations at an increased level of detail within the regions of importance enabling checks that the chemistry is integrating properly with the code and that each aligns with theoretical understanding.

8 CONCLUSION

The new web-based CCC-Tool supports users in creating continuous colormaps per CMS that have specifically optimized key colors and reference points with respect to the semantic and/or physical characteristics of target application. CCC enables identification, highlighting and isolation of specific focal areas, functionality of value to the scientists. For colormap construction, CCC enables evaluation in multiple colorspace, specifically CIE, used to test the perceptual linearity of colormaps with HSV, enabling the intuitive fine tuning of contrast levels within the colormap. The ability to interactively both manipulate and evaluate in the process of colormap construction is a capability not found in other colormap construction tools and is the feature that enable the construction of finely tuned colormaps adjusted for perceptual linearity where desired, enabled by the manipulation and evaluation in CIE spaces, in combination with the ability to maximize the internal contrast, enabled by the manipulation in HSV space. Novice users can work with the common known RGB and HSV spaces and also export colormaps using perceptual uniform colorspace for interpolation. Furthermore, predefined colormaps facilitate the creation process. With the newly developed colormap specification (CMS), it is now possible for the user to describe complex colormap structures without having to manage the large number of colors in a conventional colormap that are usually required. This prevents a overhead of processes and allows users quickly create, edit, import and export complex colormaps. By introducing middle color for some keys, we also provided a solution to a long-standing consistency problem in colormapping.

Through collaboration with domain scientist doing application oriented case studies, we owe taken design decisions. At the commonly used procedure of the placement of colors within the colormap, the inexperienced users were faced with the problem to favourable colors. Therefore, we decided on the drag and drop method of predefined structures, which were composed from experts, and if necessary own designed band structures. A desired opportunity for highlighting specific ranges, extremes or false values with a selected color, lead to allow constant bands in a CMS. A user, who is working in the field of color, opine that our modification via `Path Plot` inside the top-view-plot and the lightning-plot is advantageous to exploit the characteristics of the different color spaces and could be improved by adding the manipulation of all color space components. Therefore we extended the plots of the HSV, LAB and DIN99 version with the missing elements (e.g. Saturation and Hue for HSV). As the content in our tool continues to grow, the

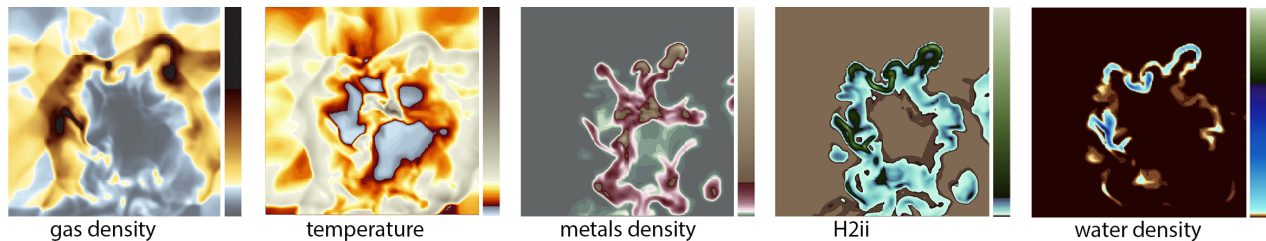


Fig. 14: Scientists examine multiple variables in order to gain an understanding into the locations and quantities where ancient water was likely to have formed. CCCTool color bar locations are crafted to highlight the data ranges for each variable that is conducive to water formation, enabling scientists to easily recognize and compare the locations over multiple variables and time steps.

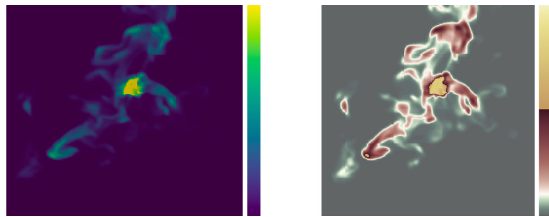


Fig. 15: Comparison of colormap impact on a slice of metals density from the supernovae simulation: left - viridis colormaps; right - cctool constructed colormap inserting the mauve color band in the data range where the metals content is sufficient for water to form.

number of opportunities overstrained non-expert users. We counteract this problem with a new design of the Edit-Section, where unnecessary content can be hidden (Fig. 2). Overall the scientists confirmed the impact and relevance of the opportunity to create application specified colormaps.

8.1 Future Work

The CCC-Tool is still in its test version, and as such, will require additions and improvements. The tool already has the capability of previewing user's data with vtk format (ASCII; Structured or Rectilinear) in real time with a colormap being edited. This feature is an early version and has to be fully developed and be stocked with build-in datasets and test-functions. Furthermore, the tool needs the utility to convert an imported colormap with hundreds of interval colors to an editable CMS with a low number of keys. For a better generation process of perceptual uniform colormaps using the CIE Lab space, we also need an algorithm for an interpolation using the metric information of CIEDE94 or CIEDE2000 [82]. Another essential future extension is a database to share CMS and enable users to use, edit, comment and rate shared CMS. This opportunity would allow a work together of the community to develop standard colormaps or to detect and improve bad CMS in the database. The additional planned future ability to equip CMS with semantics and notes can facilitate this process. For novice users, we are working on a build-in-guideline for the creation of sequential-, divergent- and multiband-colormaps based on specific tasks. With regard to the analysis, we see future work in the completion of the colormap path plots by adding 3D versions for all color spaces and a new section to support special-need users with CMS rendering in trichromacy, dichromatic, and monochromatic views. Furthermore, we plan to research the covering of transfer

functions with our tool. With the in coming opacity the development of perceptually uniform colormaps is faced with new challenges and requires new investigations.

ACKNOWLEDGMENTS

We would like to thank: William Straka III, University of Wisconsin - Madison, Space Science and Engineering Center (SSEC), Cooperative Institute for Meteorological Satellite Studies (CIMSS) for his visualizations and data of hurricane.

REFERENCES

- [1] B. E. Rogowitz and L. A. Treinish, "Data visualization: the end of the rainbow," *IEEE Spectrum*, vol. 35, no. 12, pp. 52–59, Dec 1998.
- [2] D. Borland and R. M. T. Ii, "Rainbow color map (still) considered harmful," *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 14–17, March 2007.
- [3] T.-M. Rhyne, *Applying Color Theory to Digital Media and Visualization*. CRC Press, Boca Raton, 2016.
- [4] M. Harrower and C. Brewer, "Colorbrewer.org: An online tool for selecting colour schemes for maps," *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.
- [5] R. Stauffer, G. J. Mayr, M. Dabernig, and A. Zeileis, "Somewhere over the rainbow: How to make effective use of colors in meteorological visualizations," *Bulletin of the American Meteorological Society*, vol. 96, no. 2, pp. 203–216, 2015. [Online]. Available: <https://doi.org/10.1175/BAMS-D-13-00155.1>
- [6] H. Fang, S. Walton, E. Delahaye, J. Harris, D. A. Storchak, and M. Chen, "Categorical colormap optimization with visualization case studies," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 871–880, Jan 2017.
- [7] C. C. Gramazio, D. H. Laidlaw, and K. B. Schloss, "Colorgical: Creating discriminable and preferable color palettes for information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 521–530, Jan 2017.
- [8] J. Poco, A. Mayhua, and J. Heer, "Extracting and retargeting color mappings from bitmap images of visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 637–646, Jan 2018.
- [9] D. A. Szafir, "Modeling color difference for visualization design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 392–401, Jan 2018.
- [10] R. Bujack, T. L. Turton, F. Samsel, C. Ware, D. H. Rogers, and J. Ahrens, "The good, the bad, and the ugly: A theoretical framework for the assessment of continuous colormaps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 923–933, Jan 2018.
- [11] F. Samsel, S. Klaassen, and D. H. Rogers, "Colormoves: Real-time interactive colormap construction for scientific visualization," *IEEE Computer Graphics and Applications*, vol. 38, no. 1, pp. 20–29, Jan 2018.
- [12] L. Zhou and C. Hansen, "A survey of colormaps in visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 8, pp. 2051–2069, 2016.

- [13] M. Harrower and C. A. Brewer, *ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps*. Wiley-Blackwell, 2011, ch. 3.8, pp. 261–268. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470979587.ch34>
- [14] S. Mittelstädt, D. Jäckle, F. Stoffel, and D. A. Keim, “Colorcat: Guided design of colormaps for combined analysis tasks,” in *Proc. of the Eurographics Conference on Visualization (EuroVis 2015: Short Papers)*, vol. 2, 2015.
- [15] C. Ware, *Information Visualization: Perception for Design*, 3rd ed. San Francisco, CA: Morgan Kaufman, 2012.
- [16] F. Samsel, T. Turton, C. Ware, R. Bujack, D. Rogers, G. Abram, and J. Ahrens, “Sci vis color,” accessed in March 2018. [Online]. Available: <http://sciviscolor.org/>
- [17] K. Moreland, “Color map advice for scientific visualization,” accessed in March 2018. [Online]. Available: <http://www.kennethmoreland.com/color-advice/>
- [18] P. Kovesi, “Cet perceptually uniform colour maps: Download page,” accessed in March 2018. [Online]. Available: <http://peterkovesi.com/projects/colourmaps/>
- [19] MathWorks, “Colormap editor,” accessed in March 2018. [Online]. Available: <https://uk.mathworks.com/help/matlab/ref/colormapeditor.html>
- [20] S. Silva, J. Madeira, and B. S. Santos, “There is more to color scales than meets the eye: a review on the use of color in visualization,” in *Information Visualization*, vol. 11, Jul. 2007.
- [21] S. Silva, B. S. Santos, and J. Madeira, “Using color in visualization: A survey,” *Computers & Graphics*, vol. 35, no. 2, pp. 320–333, 2011.
- [22] H. Grassmann, “Zur theorie der farbenmischung,” *Annalen der Physik*, vol. 165, no. 5, pp. 69–84, 1853.
- [23] H. Von Helmholtz, *Handbuch der physiologischen Optik*. Voss, 1867, vol. 9.
- [24] J. Guild, “The colorimetric properties of the spectrum,” *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 230, pp. 149–187, 1932.
- [25] I. C. on Illumination, *Colorimetry*, ser. CIE technical report. Commission internationale de l’Eclairage, CIE Central Bureau, 2004. [Online]. Available: <https://books.google.com/books?id=PiNkAAAACAAJ>
- [26] H. S. Fairman, M. H. Brill, H. Hemmendinger *et al.*, “How the cie 1931 color-matching functions were derived from wright-guild data,” *Color Research & Application*, vol. 22, no. 1, pp. 11–23, 1997.
- [27] A. Broadbent, “Calculation from the original experimental data of the cie 1931 rgb standard observer spectral chromaticity coordinates and color matching functions,” *Québec, Canada: Département de génie chimique, Université de Sherbrooke*, 2008.
- [28] H. Buring, “Eigenschaften des farbenraumes nach din 6176 (din99-formel) und seine bedeutung für die industrielle anwendung,” in *8. Workshop Farbbildverarbeitung der German Color Group*, 2002, pp. 11–17.
- [29] M. R. Luo, G. Cui, and C. Li, “Uniform colour spaces based on ciec02 colour appearance model,” *Color Research & Application*, vol. 31, no. 4, pp. 320–330, 2006.
- [30] D. B. Judd, “Ideal color space,” *Color Engineering*, vol. 8, no. 2, p. 37, 1970.
- [31] D. L. MacAdam, “Visual sensitivities to color differences in daylight*,” *J. Opt. Soc. Am.*, vol. 32, no. 5, pp. 247–274, May 1942. [Online]. Available: <http://www.osapublishing.org/abstract.cfm?URI=josa-32-5-247>
- [32] C. E. Helm, “Multidimensional ratio scaling analysis of perceived color relations,” *JOSA*, vol. 54, no. 2, pp. 256–262, 1964.
- [33] D. Nickerson and K. F. Stultz, “Color tolerance specification,” *JOSA*, vol. 34, no. 9, pp. 550–570, 1944.
- [34] M. Mahy, L. Eycken, and A. Oosterlinck, “Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV,” *Color Research & Application*, vol. 19, no. 2, pp. 105–121, 1994.
- [35] M. R. Luo, G. Cui, and B. Rigg, “The development of the cie 2000 colour-difference formula: Ciede2000,” *Color Research & Application*, vol. 26, no. 5, pp. 340–350, 2001.
- [36] R. Huertas, M. Melgosa, and C. Oleari, “Performance of a color-difference formula based on OSA-UCS space using small-medium color differences,” *JOSA A*, vol. 23, no. 9, pp. 2077–2084, 2006.
- [37] P. K. Robertson and J. F. O’Callaghan, “The generation of color sequences for univariate and bivariate mapping,” *IEEE Computer Graphics and Applications*, vol. 6, no. 2, pp. 24–32, 1986.
- [38] B. Pham, “Spline-based color sequences for univariate, bivariate and trivariate mapping,” in *Proceedings of the 1st conference on Visualization’90*. IEEE Computer Society Press, 1990, pp. 202–208.
- [39] H. Levkowitz and G. T. Herman, “The design and evaluation of color scales for image data,” *IEEE Computer Graphics and Applications*, vol. 12, no. 1, pp. 72–80, 1992.
- [40] H. Levkowitz, “Perceptual steps along color scales,” *International Journal of Imaging Systems and Technology*, vol. 7, no. 2, pp. 97–101, 1996.
- [41] H. Zhang and E. Montag, “Perceptual color scales for univariate and bivariate data display,” *The Society for Imaging Science and Technology (IS&T)*, 2006.
- [42] A. Zeileis, K. Hornik, and P. Murrell, “Escaping RGBland: selecting colors for statistical graphics,” *Computational Statistics & Data Analysis*, vol. 53, no. 9, pp. 3259–3270, 2009.
- [43] N. Silvestrini and E. Fischer, “Idee farbe,” *Farbsysteme in Kunst und Wissenschaft*. Baurmann & Stromer, Zürich, 1994.
- [44] K. R. Sloan and C. M. Brown, “Color map techniques,” *Computer Graphics and Image Processing*, vol. 10, no. 4, pp. 297–317, 1979.
- [45] B. E. Trumbo, “A theory for coloring bivariate statistical maps,” *The American Statistician*, vol. 35, no. 4, pp. 220–226, 1981.
- [46] S. M. Pizer, J. B. Zimmerman, and R. E. Johnston, “Contrast transmission in medical image display,” in *1st International Symposium on Medical Imaging and Image Interpretation*. International Society for Optics and Photonics, 1982, pp. 2–9.
- [47] J. Tajima, “Uniform color scale applications to computer graphics,” *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 3, pp. 305–325, 1983.
- [48] K. Moreland, “Diverging color maps for scientific visualization,” in *International Symposium on Visual Computing*. Springer, 2009, pp. 92–103.
- [49] C. Tominski, G. Fuchs, and H. Schumann, “Task-driven color coding,” in *2008 12th International Conference Information Visualisation*. IEEE, 2008, pp. 373–380.
- [50] M. Wijffelaars, R. Vliegen, J. J. Van Wijk, and E.-J. Van Der Linden, “Generating color palettes using intuitive parameters,” *Computer Graphics Forum*, vol. 27, no. 3, pp. 743–750, 2008. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2008.01203.x>
- [51] D. L. Gresh, “Self-corrected perceptual colormaps,” IBM, Tech. Rep., 2008, RC24542 (W0804-104).
- [52] C. Ware, T. L. Turton, F. Samsel, R. Bujack, and D. H. Rogers, “Evaluating the Perceptual Uniformity of Color Sequences for Feature Discrimination,” in *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)*, K. Lawonn, N. Smit, and D. Cunningham, Eds. The Eurographics Association, 2017, pp. 7–11.
- [53] H. Wainer and C. M. Francolini, “An empirical inquiry concerning human understanding of two-variable color maps,” *The American Statistician*, vol. 34, no. 2, pp. 81–93, 1980.
- [54] A. Light and P. J. Bartlein, “The end of the rainbow? Color schemes for improved data graphics,” *Eos*, vol. 85, no. 40, pp. 385–391, 2004.
- [55] M. Eisemann, G. Albuquerque, and M. Magnor, “Data driven color mapping,” in *Proceedings of EuroVA: International Workshop on Visual Analytics, Bergen, Norway*. Citeseer, 2011.
- [56] C. Ware, “Color sequences for univariate maps: Theory, experiments and principles,” *IEEE Computer Graphics and Applications*, vol. 8, no. 5, pp. 41–49, 1988.
- [57] L. D. Bergman, B. E. Rogowitz, and L. A. Treinish, “A rule-based tool for assisting colormap selection,” in *Proceedings of the 6th conference on Visualization’95*. IEEE Computer Society, 1995, p. 118.
- [58] B. E. Rogowitz, L. A. Treinish, and S. Bryson, “How not to lie with visualization,” *Computers in Physics*, vol. 10, no. 3, pp. 268–273, 1996.
- [59] P. L. Rheingans, “Task-based color scale design,” in *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*. International Society for Optics and Photonics, 2000, pp. 35–43.
- [60] B. E. Rogowitz and A. D. Kalvin, “The ‘Which Blair Project’: a quick visual method for evaluating perceptual color maps,” in *Visualization, 2001. VIS’01. Proceedings*. IEEE, 2001, pp. 183–556.
- [61] C. Brewer, *Designing Better Maps: A Guide for Gis Users*. Environmental Systems Research, 2004.
- [62] S. H. Pizer, J. B. Zimmerman, and R. E. Johnston, “Concepts of the display of medical images,” *IEEE Transactions on Nuclear Science*, vol. 29, no. 4, pp. 1322–1330, 1982.

- [63] D. Borland and A. Huber, "Collaboration-specific color-map design," *IEEE Computer Graphics and Applications*, vol. 31, no. 4, pp. 7–11, July 2011.
- [64] F. Samsel, S. Klaassen, M. Petersen, T. L. Turton, G. Abram, D. H. Rogers, and J. Ahrens, "Interactive colormapping: Enabling multiple data range and detailed views of ocean salinity," in *Proc. ACM CHI (Extended Abstracts)*, 2016, pp. 700–709.
- [65] M. Green, "Toward a perceptual science of multidimensional data visualization: Bertin and beyond," *ERGO/GERO Human Factors Science*, vol. 8, 1998.
- [66] J. Ahrens, B. Geveci, and C. Law, "36 - paraview: An end-user tool for large-data visualization," in *Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Burlington: Butterworth-Heinemann, 2005, pp. 717 – 731. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780123875822500381>
- [67] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, and P. Navrátil, "VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data," in *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, Oct 2012, pp. 357–372.
- [68] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [69] R. Borgo, K. Proctor, M. Chen, H. Jänicke, T. Murray, and I. M. Thornton, "Evaluating the impact of task demands and block resolution on the effectiveness of pixel-based visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 963–972, 2010.
- [70] F. Samsel, M. Petersen, T. Geld, G. Abram, J. Wendelberger, and J. Ahrens, "Colormaps that improve perception of high-resolution ocean data," in *Proc. ACM CHI (Extended Abstracts)*, 2015, pp. 703–710.
- [71] F. Samsel, T. L. Turton, P. Wolfram, and R. Bujack, "Intuitive Colormaps for Environmental Visualization," in *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, K. Rink, A. Middel, D. Zeckzer, and R. Bujack, Eds. The Eurographics Association, 2017.
- [72] F. Samsel, J. M. Patchett, D. H. Rogers, and K. Tsai, "Employing color theory to visualize volume-rendered multivariate ensembles of asteroid impact simulations," in *Proc. ACM CHI (Extended Abstracts)*, 2016, pp. 1126–1134.
- [73] B. R. G. R. G. Cui, M. R. Luo and K. Witt, "Uniform colour spaces based on the din99 colour-difference formula," *Color Research and Application*, vol. 27, pp. 282–290, 2002.
- [74] J. M. P. Yong-Keun Lee, "Comparison of cie lab, ciede 2000, and din 99 color differences between various shades of resin composites," *International Journal of Prosthodontics*, vol. 18, pp. 150–155, 2005.
- [75] J. Itten and F. Birren, *The Elements of Color: A Treatise on the Color System of Johannes Itten Based on His Book the Art of Color*, 1st ed. New York, NY: Van Nostrand Reinhold Company, 1970.
- [76] J. Albers, *The Interaction of Color*. New Haven, CT: Yale University Press, 2009.
- [77] R. Bujack, T. L. Turton, D. Rogers, and J. Ahrens, "Ordering Perceptions about Perceptual Order," in *IEEE Scientific Visualization Conference (SciVis) Short Papers*. IEEE, 2018.
- [78] P. Nardini, M. Böttinger, G. Scheuermann, and M. Schmidt, "Visual Study of the Benguela Upwelling System using Pathline Predicates," in *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, K. Rink, A. Middel, D. Zeckzer, and R. Bujack, Eds. The Eurographics Association, 2017.
- [79] M. Carr, "Estimation of potential productivity in eastern boundary currents using remote sensing," *Deep Sea Research Part II*, vol. 49, pp. 59–80, 2002.
- [80] F. P. Chavez and M. Messié, "A comparison of eastern boundary upwelling ecosystems," *Progress in Oceanography*, vol. 83, no. 1–4, pp. 80–96, Dec 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.pocean.2009.07.032>
- [81] B. Wiggins, a. K. H. F. Samsel, G. Abram, S. Jones, a. M. T. A. Gagliano, and J. Smidt, "The first water in the universe," *Parallel Computing*, 2018.
- [82] M. Zeyen, T. Post, a. A. Hans Hagen, D. Rogers, and R. Bujack, "Color Interpolation for Non-Euclidean Color Spaces," in *IEEE Scientific Visualization Conference (SciVis) Short Papers*. IEEE, 2018.

Pascal Nardini received a master degree in computer science in 2017 from the Leipzig University, after which he started working on his Ph. D. in computer science in the Image and Signal Processing Group at Leipzig University.

Min Chen received his PhD degree from University of Wales in 1991. He is currently a professor of scientific visualization at Oxford University and a fellow of Pembroke College. Before joining Oxford, he held research and faculty positions at Swansea University. His research interests include visualization, computer graphics and human-computer interaction. His services to the research community include papers co-chair of IEEE Visualization 2007 and 2008, IEEE VAST 2014 and 2015, and Eurographics 2011; co-chair of Volume Graphics 1999 and 2006, and EuroVis 2014; associate editor-in-chief of IEEE TVCG; editor-in-chief of Computer Graphics Forum; and co-director of Wales Research Institute of Visual Computing. He is a fellow of BCS, EG and LSW.

Francesca Samsel Francesca Samsel, a research associate at University of Texas at Austin, works extensively with the Data of Science Scale team and research scientists at Los Alamos National Laboratory. She holds an MFA from the University of Washington, using her artistic expertise and color strategies to assist scientists in extracting more understanding from their data sets, (SciVisColor.org). Working with public institutions, she is involved in developing science communication through the languages of art.

Roxana Bujack graduated as M.Sc. in mathematics at Leipzig University in Germany in 2010. Then, she received her B.Sc. and her Ph.D. in computer science in the Image and Signal Processing Group at Leipzig University. She worked as a postdoctoral researcher at IDAV at the University of California, Davis in the USA and at the Computer Graphics and HCI Group at the Technical University Kaiserslautern. Currently, she is a staff scientist in the Data Science at Scale Team at Los Alamos National Laboratory. Her research interests include data science, visualization, pattern detection, vector fields, Lagrangian flow representations, moment invariants, and color theory.

Michael Böttinger received his Diploma in Geophysics from the University of Hamburg, Germany, in 1988, after which he started working in the field of climate modeling at the Max Planck Institute for Meteorology. In 1990 he joined the scientific visualization team at the German Climate Computing Center (DKRZ). Today, he leads DKRZ's visualization and public relations group. His research is application oriented and focuses on scientific visualization of climate model data for scientific discovery as well as for communication to the broad public.

Gerik Scheuermann received a master degree (diplom) in mathematics in 1995 and a PhD degree in computer science in 1999, both from the Technical University of Kaiserslautern. He is a full professor at the University of Leipzig since 2004. He has coauthored more than 200 reviewed book chapters, journal articles, and conference papers. His current research interests focus on visualization and visual analytics, especially on feature and topology-based methods, flow visualization, tensor visualization, environmental visualization, document visualization and visualization for life sciences. He has served as paper cochair for Eurovis 2008, IEEE SciVis 2011, IEEE SciVis 2012, and IEEE PacificVis 2015. He has organized TopoInVis 2007, AGACSE 2008 and EuroVis 2013, as well as two Dagstuhl Seminars on Visualization. Furthermore, we was associated editor of IEEE TVCG from 2008-2012, and serves currently as associated editor for IEEE CG&A, The Visual Computer and Computer Graphics Forum.