LA-UR-20-22115
Approved for public release; distribution is unlimited.

Title: A Testing Environment for Continuous Colormaps

Author(s): Nardini, Pascal Antonio
Chen, Min
Bujack, Roxana Barbara
Boettinger, Michael
Scheuermann, Gerik

Intended for: IEEE VIs, 2020-10-25 (Salt Lake City, Utah, United States)

Issued: 2021-08-18 (rev.1)

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

A Testing Environment for Continuous Colormaps

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

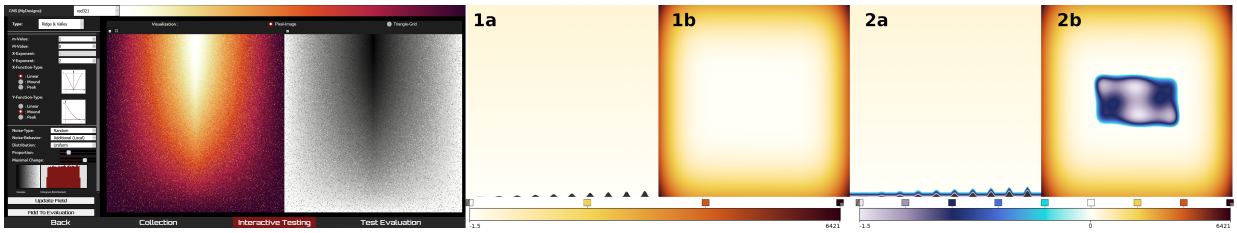

Fig. 1. **Left**: Screenshot of our new interactive test suite implemented in the CCC-Tool. **Right**: Two visualizations show at **b** the valley shaped Six-Hump Camel Function [15] for the area $[-5,5] \times [-5,5]$ and at **a** the *LittleBit* test function (Sect. 4.2.2). Special features of the Six-Hump Camel function are irregularly shaped troughs in the center of the area that are more than two orders smaller than the data range difference between center and boundary. The uniformly spaced colormap **1** cannot reveal the troughs, while the last nonlinear colormap **2** clearly shows the topological structure of the irregularities in the center of the domain.

Abstract— Many computer science disciplines (e.g., combinatorial optimization, natural language processing, and information retrieval) use standard or established test suites for evaluating algorithms. In visualization, similar approaches have been adopted in some areas (e.g., volume visualization), while user testimonies and empirical studies have been the dominant means of evaluation in most other areas, such as designing colormaps. In this paper, we propose to establish a test suite for evaluating the design of colormaps. With such a suite, the users can observe the effects when different continuous colormaps are applied to planar scalar fields that may exhibit various characteristic features, such as jumps, local extrema, ridge or valley lines, different distributions of scalar values, different gradients, different signal frequencies, different levels of noise, and so on. The suite also includes an expansible collection of real-world data sets including the most popular data for colormap testing in the visualization literature. The test suite has been integrated into a web-based application for creating continuous colormaps (https://ccctool.com/), facilitating close inter-operation between design and evaluation processes. This new facility complements traditional evaluation methods such as user testimonies and empirical studies.

Index Terms—Testing Environment, Color Perception, Scalar Analysis

1 Introduction

In many fields in computer science, algorithms are commonly evaluated using open libraries of predefined tests. A typical example is the field of combinatorial optimization with ACM GECCO [1] as its leading conference. New algorithms are often tested against standard problems (e.g., the traveling salesman problem or the quadratic assignment problem), using established libraries (e.g., SPlib [35] or QAPlib [11]). These libraries contain test data sets with increasing difficulty, each typically focusing on different (known) challenges in combinatorial optimization. Similarly, the natural language processing (NLP) community frequently uses test data sets in the Semantic Evaluation suite [63]. In the past, algorithmic developments in NLP benefited substantially from the TREC collection of test data [55].

Algorithmic development has been a core component of scientific visualization. There were earlier attempts to create an open collection of test data, such as in the area of volume visualization, and no many collections are still remaining (e.g., Klacansky:2020:web). Whilst there are data sets from the SciVis contests, the evaluation of an algorithm or the comparison of different algorithms is usually conducted with proprietary data sets chosen by the evaluator (who is often also the algorithm developer). In some cases, the evaluation or comparison is accompanied by a usability study involving domain experts. A colormap implicitly encodes an algorithmic transformation from a data set to its visual representation. Hence the evaluation of a transformation is primarily about the colormap concerned. In this paper, we report a test suite for evaluating continuous colormaps, in the spirit of the open approach seen in other fields as well as in the earlier decades of scientific visualization. This does not mean to abandon usability studies, but the test suite can efficiently answer many questions and help improve colormap designs before or after a usability study. The test suite will also allow the VIS community to collect test data representing different challenges to colormap designs or different applications as well as helping identifying guidelines to use or pitfalls to avoid.

Color mapping is probably the most commonly used method for transforming data variables into visual variables. Color heatmaps are ubiquitous in all scientific applications featuring captured or simulated data with a spatial context (e.g., geographical data, imagery data, simulation results underpinned by some geometry, and so on). Every domain expert dealing with 2D scalar fields knows how to observe color heatmap. Color mapping is also an integral component in almost all complex rendering techniques for visualizing height-, scalar-, vector-, and tensor-fields. One would expect to find some color mapping methods in any free and commercial visualization software system [2,12,52].

Selecting a good (continuous) colormap has been an enduring topic in visualization for many years (e.g., [23,42]). There are several metrics for measuring the quality of colormaps (e.g., perceived color differences [10,44]) and tools for creating colormaps (e.g., [30]). However, there is yet a test suite for colormap designers to try out some design options in order to investigate how they may reveal patterns in different

- *Pascal Nardini is with Institute of Computer Science, University of Leipzig, Germany. E-mail: 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.*
- *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 xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

test data sets. Although one may test a colormap using an existing application data set, this is usually not adequate enough because given a context, a color map should not only reveal important patterns in known data sets, but also do the same for new data sets that may arrive in the future. A test suite can enable designers to speculate on different "what-if" scenarios, and examine how a colormap may react.

This paper presents a test suite designed to serve the above purpose. The test suite consists of three parts. The first part contains a set of test functions that provide a comprehensive set of potential *local properties* of 2D scalar functions. For example, some functions may feature "jumps" at different value levels and with different increments or decrements, gradients of different scales starting from different levels, critical points, ridges, and different frequencies. In Sect. 4, we give a mathematical reasoning behind the selection of these functions.

The second part of the test suite contains functions to model a set of *global properties*. The designer of a colormap often encounters challenging data sets with complex variations of data ranges in different regions. For example, a multi-band colormap may suit a region with large but smooth jumps, while a slow-paced sequential colormap may suit a region with many small but volatile jumps. Hence test functions that simulate such complex variations challenge colormap designers as well as colormap optimization algorithms. We use some well-known sample functions from the (continuous) optimization literature for this purpose. Complex variations can also be caused by some local properties alone, such as different signal-to-noise levels.

In addition to the two categories of test functions, the third part provides *real-world data sets* collected from different applications. We expect that this part of the test suite will continue to expend.

By integrating the test suite in our open-access CCC-Tool, which enables users to create, edit and analyze colormaps [30], we now provide a complete design and test environment for developing and testing of application-specific colormaps and for supporting all types of visualization applications and domain-specific data sets.

2 Related Work

2.1 Color Mapping

Continuous color mapping (also heat mapping) refers to the association of a color to a scalar value over a domain and can be considered the most popular visualization technique for two-dimensional scalar data.

There are many heuristic rules for designing a good colormap, which have been applied mostly intuitively for hundreds of years [49, 50, 66]. The most important ones are order, high discriminative power, uniformity, and smoothness [10].

While some colormaps have been designed to sever as default colormaps for many data sets and can perform reasonably well in terms of rule compliance [29], many colormaps are purposely-designed according to application-specific requirements such as the shape of the data, the audience, the display, or the visualization goal [3, 5, 36, 51]. The number of possible colormap configurations and the body of related work on this topic are huge [3, 36, 43, 57, 60].

An effort has been made to measure the quality of colormaps with respect to these rules quantitatively [10, 21, 29, 37, 54] or experimentally [19, 39, 60, 61], in order to develop theories and algorithms that can help automate the generation, evaluation, and improvement of colormaps. Although such theories and algorithms are usually general enough to be application-independent, the design of colormaps in many practical applications can only be effective if one includes some application-specific semantics in the design, such as key values, critical ranges, non-linearity, probability distribution of certain values or certain spatial relationships among values, and so on. Supporting such application-specific design effort is the goal of this work.

2.2 Colormap Test Data

So far, there is no test suite for colormaps. However, the literature has provided various examples where some data sets were used for comparing color maps and demonstrating color mapping algorithms.

Sloan and Brown [51] suggest treating a colormap as a path through a color space and stress that the optimal choice of a colormap depends on the task, human perception, and display. They showcase their findings with *x*-ray and satellite images. Wainer and Francolini [58] point out the importance of order in a visual variable using statistical information on maps. Pizer [33, 34] stresses the importance of uniformity in a colormap, of which the curve of just noticeable differences (JNDs) is constant and in a natural order. The uniformity can be achieved by increasing monotonically in brightness or each of the RGB components, such that the order of their intensities does not change throughout the colormap. Tajima [54] uses colormaps with regular color differences in a perceptually uniform colorspace to achieve perceptually uniform color mapping of satellite images. Levkowitz and Herman [21] suggest an algorithm for creating colormaps that produces maximal color differences while satisfying monotonicity in RGB, hue, saturation, and brightness. They test them with medical data from CT scans. Bernard et al. [4] suggest definitions of colormap properties and build relations to mathematical criteria for their assessment and map them to different tasks independent from data properties in the context of bivariate colormaps. They test the criteria on analytical data that has different shapes (e.g., different gradients and spherical surfaces).

Pizer states that the qualitative task is more important in color mapping applications, because quantitative tasks can be better performed using contours or by explicitly displaying the value when the user hovers over a data point with the mouse. He uses medical images, including CT scans and digital subtraction fluorography, as example data sets. Ware [60] also distinguishes qualitative and quantitative tasks. He agrees that the qualitative characteristics are more important and explicitly mentions tables as a suitable means for the visualization of quantitative values. On the one hand, he finds that monotonic change in luminance is important to see the overall form (qualitative) of his analytic test data consisting of linear gradients, ridges, convexity, concavity, saddles, discontinuities, and cusps. On the other hand, his experiments show that when a colormap consists of only one completely monotonic path in a single perceptual channel, the quantitative task is error-prone if one tries to read the exact data values based on the visualization.

Rogowitz, et al. [3, 19, 38, 40–43] distinguish different tasks (isomorphic, segmentation, and highlighting), data types, (nominal, ordinal, interval, and ratio), and spatial frequency (low, high), recommending colormap properties for each combination. They perform experiments on the visual perception of contrast in colormaps using Gaussian or Gabor targets of varying strength superimposed on linear gradients of common colormaps [19, 39]. Rogowitz et al. use a huge variety of data through their extensive experiments, for example, text [40], MRI scans of the human head [42, 42], weather data showing clouds or ozone distribution [42, 42], vector field data from a simulation of the earth's magnetic field or jet flows [3, 42], measurements from remote sensing [43], cartographic height data [42], analytic data covering a broad spectrum of frequencies such as planar wave patterns [43], linear gradients distorted by a Gaussian or Gabor target of increasing magnitude [19, 39], the luminance of a human face photograph [38], and so on. Their work demonstrates the diversity in the application field of color mapping and how important it is for a colormap to encode application-specific semantics. Zhang and Montag [65] evaluate the quality of colormaps designed in a uniform color space with a user study using a CAT scan and scientific measurements such as remote sensing and topographic height data. Gresh [16] measures the JND between colors in a colormap, using cartographic height data. Ware et al. [61] generate stimuli for experiments on colormap uniformity by superimposing vertical strips of Gabor filters of different spatial extent over popular colormaps with magnitudes ranging from nonexistence on the top to very strong contrast on the bottom. The users' task is to pick the location where they could first perceive the distortion.

Light and Bartlein [22] warn of using the rainbow colormap, showing that it is highly confusing for color vision impaired users at the example of temperature data covering North and South America. Borland [6] also criticizes the rainbow colormap for its lack of order. He compares different colormaps based on analytic test data that features a spectrum of changing frequencies, different surface shapes, and gradients. Kindlmann et al. [20] suggest a method to evaluate users' perception of luminance using a photograph of a human face. Schulze-Wollgast et

Table 1. The most popular test data for colormap testing in the visualization literature.

analytic data	[6, 19, 27, 29, 39, 43, 56, 59, 60]
	[4, 26, 61]
statistics and maps	[8, 9, 13, 14, 27, 47, 51, 57, 64]
medical imaging	[21, 33, 34, 42, 51, 59, 65]
scientific measurements	[14, 16, 22, 29, 30, 42, 43, 46, 64, 65]
scientific simulations	[3, 5, 29, 30, 42, 45, 46, 56]
photographs	[20, 38, 51, 54]

al. [47] focus on the task of comparing data using statistical information on maps. Tominski et al. [57] also stress that the characteristics of the data, tasks, goals, user, and output device need to be taken into account. They introduce their task-color-cube, which gives recommendations for the different cases. They use cartographic data to demonstrate their findings. Wang [59] chooses color for illustrative visualizations using medical data and measurements of transmission electron microscopy (TEM), analytic jumps, and mixing of rectangles. Zeileis et al. [64] provide code to generate color palettes in the cylindrical coordinates of CIEUV and showcase results using geyser eruption data of Old Faithful and cartographic data. Moreland [29] presents an algorithm that generates diverging colormaps that have a long path through CIELAB without sudden non-smooth bends. His red-blue diverging colormap is the current default in ParaView [2]. He tests different colormaps with data representing a spectrum of frequencies and gradients partly distorted by noise. He also stresses the importance of testing on 3D surfaces where shading and color mapping compete, e.g., the density on the surface of objects in flow simulation data or on 3D renderings of cartographic height data. Borland [5] collaborates with an application scientist working on urban airflow. They suggest combining existing colormaps to design domain-specific ones, and in case of doubt stick with the black-body radiation map. They sacrifice traditional rules (e.g., order) to satisfy the needs (huge discriminative power) of the application. Eisemann et al. [13] separate the adaption of the histogram of the data from the color mapping task, introducing an interactive pre-colormapping transformation for statistical information on maps. Thompson et al. [56] suggest applying special colors outside the usual gradient of the colormap to dominantly-occurring values, which are "prominent" values occurring with high frequency. Their test data includes the analytic Mandelbrot fractal and flow simulation results, which are partly provided as examples in ParaView. Brewer [8, 9] provides an online tool to choose carefully designed discrete colormaps. This is perhaps the most widely used tool for discrete colormaps. Mittelstädt et al. [26,27] present a tool that helps to find a suitable colormap for different task combinations. They showcase their findings with analytical data, like gradients and jumps, and real-world maps. Samsel et al. [45,46] provide intuitive colormaps designed by an artist to visualize ocean simulations and scientific measurements in the environmental sciences. Fang et al. [14] present an optimization tool for categorical colormaps, and use the tool to improve the colormap of the London underground map and that for seismological data visualization. Nardini et al. [30] provides an online tool, the `CCC-Tool`, for creating, editing, and analyzing continuous colormaps, demonstrating its uses with captured hurricane data, simulated ocean temperature data, and results of simulating ancient water formation.

All in all, we found that the most popular way of evaluating the quality of colormaps in the literature is the use of specifically designed analytic data like gradients, ridges, different surface shapes, fractals, jumps, or different frequencies, because these synthetic data sets help to identify specific properties of the colormaps. The second most common use is cartographic maps, which reflects the historical use of color mapping. Furthermore, it is also common to use data in typical applications of scientific visualization as test data, e.g., fluid simulations (wind, ocean, turbulence), scientific measurements (weather, clouds, chemical concentration, temperature, elevation data), and medical imaging (x-ray, CT scan, digital subtraction fluorography, transmission electron microscopy). A summary can be found in Table 1. We have carefully designed our colormap test suite according to these findings, not only providing an extensive selection of expressive analytic data, but also containing real-world data from different scientific applications.

3 Motivation

In several fields of computer science, the use of established test suites for evaluating techniques is standard or commonplace. The motivation for this paper is to introduce such a test suite to scientific visualization. So far, user testimonies and empirical studies have been the dominant means of evaluation in the literature. With this work, we would like to initiate the development of an open resource that the community can use to conduct extensive and rigorous tests on various colormap designs. We also anticipate that the community will contribute new tests to this resource continuously, including but not limited to tests for colormaps used in vector- or tensor-field visualization. Such a test suite can also provide user-centered evaluation methods with stimuli and case studies, while stimulating new hypotheses to be investigated using perceptual and cognitive experiments.

The development of testing functions in this paper deals with the common features that pose challenges in scalar analysis, such as jumps, local extrema, ridge or valley lines, different distributions of scalar values, different gradients, different signal frequencies, different levels of noise, and so on. This scope should be extended in future work progressively with more and more complex or specialized cases.

The main design goal of our test suite is to provide a set of intuitive functions, each of which deals with one particular challenge at a time. They should be easy to interpret and to customize by experts as well as non-expert users. This aspired simplicity in design can be exploited in future work to facilitate automatic production of test reports or automatic optimization of colormaps with respect to a selection of tests.

At present, this initial development should provides a set of test functions simulating a variety of planar scalar fields with different characteristic features. It should enable the users to observe the effects when different continuous colormaps are applied to scalar fields that have the characteristic features similar to those featured in an application. In many situations, the users may anticipate certain features in data sets that are yet to arrive, and would like to ensure that the color mapping can reveal such features effectively when the data arrives. Finding and defining a suitable testing function is usually easier than manually creating a data set. Especially, unlike a synthetic data set, a test function is normally resolution-independent and is accompanied by some parameters for customization.

In addition, the test suite should provide users with data sets that come from real-world applications, possibly with some modification wherever appropriate. Such an application-focused collection can be compiled from the most popular data for colormap testing in the visualization literature. Since both the collection of test functions and that of real-world data sets are extensible, the field of visualization may soon see a powerful test suite for evaluating colormap design. This is desirably in line with other computer science disciplines.

4 Test Suite

The first design goal of our test functions is to allow intuitive interpretation of colormap properties by users. This requires each test function to have an easily-understandable behavior, and to have a clear mathematical description that can be reproduced consistently across different implementation. The second design goal is to build the collection of test functions on the existing analytic examples in the literature surveyed in Sect. 2 to ensure that the existing experiments can be repeated and compared with new experiments. The third design goal is to help users to find the test suite and to conduct tests easily. Hence we integrate the test suite with the `CCC-Tool`, allowing users to conduct tests immediately after creating or editing a colormap specification.

As mentioned in Sect. 1, our test suite has three parts: local tests, global tests, and a set of application data sets. The local tests are mostly based on analytic examples in the literature and are defined with considerations from calculus to cover most local properties of scalar functions. The global tests feature analytic properties of scalar functions that are not local, such as signal to noise ratio, global topological properties, different levels of variation, etc. Finally, the application-

a_0	a_3	a_1	a_3	a_2	a_3	a_3	a_3
a_0	a_2	a_1	a_2	a_2	a_2	a_3	a_2
a_0	a_1	a_1	a_1	a_2	a_1	a_3	a_1
a_0	a_0	a_1	a_0	a_2	a_0	a_3	a_0

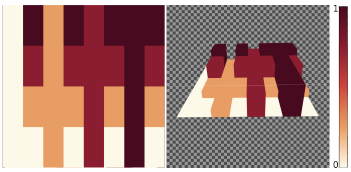

Fig. 2. **Left:** The table shows the structure of the neighborhood with four elements ($A = \{a_0, a_1, a_2, a_3\}$). The odd indexed columns (yellow) always include the same value, increasing from the first to the last column. The even indexed columns (orange) contain the whole set of test values in increasing order. **Right:** Neighborhood variation test with $A = \{0.0, 0.25, 0.75, 1.0\}$ for the colormap displayed below including a three-dimensional version encoding the values through height.

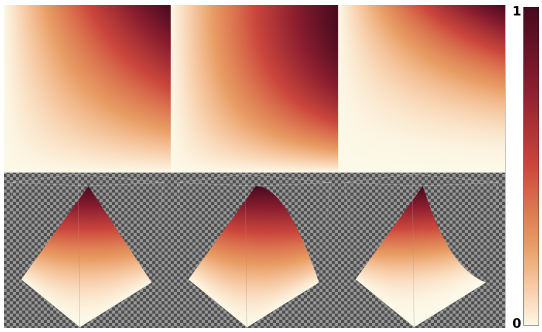

Fig. 3. Three gradient tests, with $r = 0$, $R = 1.0$. **Top Row**: Color mapping visualizations of the `Gradient Variation` function for the types *linear*, *convex*, and *concave* with $T_x = T_y$ and $b = 1$ for the first type and $b = 2$ for the other types. **Bottom Row**: 3D height-map visualizations of the three gradient tests.

specific data sets reflect the well-documented fact that colormaps should also be evaluated using real-world data sets.

The mathematical notions in this section use a few common parameters. The user-defined parameters, m, M, set the test function range, with $r, R \in \mathbb{R} \wedge r \neq R$. R and r determine the minimum m and maximum M of the test function with $m < M \in \mathbb{R}$. With $b \in \mathbb{N}$ the user can select an exponent that describes the polynomial order. For functions with enumerated cases, the user can select a specific option T.

4.1 Local Tests

Our basic design principle behind the local tests is classical calculus. Local means that these test functions help to check the appearance of local properties of a scalar function after mapping it to color with the selected color map. The main idea is to use typical local approximations like low order Taylor series expansions to create the test functions. We use step functions to show the effect of discontinuities, and provide functions with different gradients, various local extrema, saddles, ridges, and valley lines. This corresponds to ideas in the literature, as shown in Sect. 2, e.g., works by Mittelstädt [26, 27] or Ware [60]. We also use elements of Fourier calculus by providing functions to test the effect of different frequencies. The final test looks at the colormap's potential to visually reveal small differences within the data range, which might be an important colormap design goal.

4.1.1 Step Functions

Some popular test images in the literature use steps between adjacent pixels [26, 27, 59]. In terms of calculus, this means to use a function with discontinuities. Ideally, the function should have different step heights starting from different levels. For this purpose, we define a set $A = a_0, \ldots, a_{n-1}$ of increasing test values $a_i < a_{i+1}$. The function is split into a rectangular grid with constant values. Uneven columns contain in ascending sequence one value of A, while even ones contain A in increasing order. This means that some steps appear multiple times,

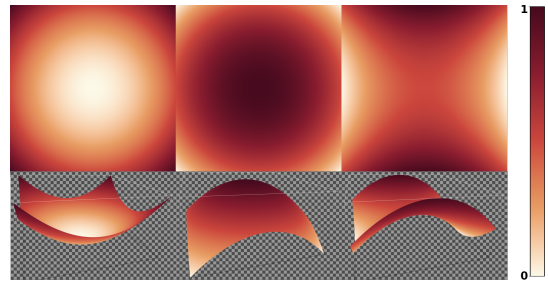

Fig. 4. This figure shows a 2D color mapping representation (top) and a 3D height-map (bottom) of the 2d scalar fields created with the test function yielding a minimum with $o = 1$ and $p = 1$ (left), a maximum with $o = -1$ and $p = -1$ (middle), and a saddle with $o = -1$ and $p = 1$ (right).

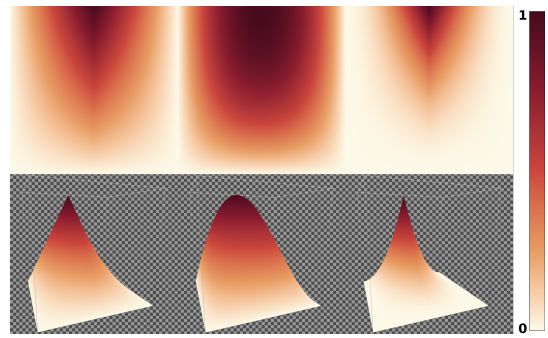

Fig. 5. Three ridge/valley-line tests (columns), with $r = 0$, $R = 1.0$. The ridge/valley-line is always centrally at $x = 0$. **Top Row**: Color mapping with $T_x = T_y = linear$, $T_x = T_y = concave$, and $T_x = T_y = convex$, and $b = 2$ in the latter two cases. **Bottom Row**: 3D height-map versions of the same tests.

but the function is rather simple to remember; see Fig. 2. Formally, the function $f_{Step} : [0, 2n) \times [0, n) \to \mathbb{R}$ is given by

$$f_{Step}(x, y) = \begin{cases} a_{\lfloor x \rfloor / 2} & \textbf{if } \lfloor x \rfloor \mod 2 = 0 \\ a_{\lfloor y \rfloor} & \textbf{otherwise} \end{cases} \tag{1}$$

where $\lfloor x \rfloor$ denotes the largest integer smaller or equal to x. An observer can quickly identify the column, including one concrete test-value, and compare it with all values using the adjacent column.

4.1.2 Gradient Variation

In the literature, the most common analytic example features gradient variation [4, 6, 26, 27, 29]. In terms of calculus, at first glance, this may suggest using a first-order Taylor approximation. However, the work by Ware [60] indicates that one should also test concave or convex properties, which means to use polynomials of somewhat higher-order, e.g., quadratic polynomials.

For our `Gradient Variation` test, we defined a test function $f_{Grad}(x, y) : [0, 1] \times [0, 1] \to \mathbb{R}$ to examine different gradients including a concave or convex pattern. The two options T_x, T_y determine if the behavior is convex or concave along the x-axis or the y-axis.

Basically, along each horizontal line, we start at revise$r$ and interpolate to some value $g(y) \in \mathbb{R}$. The function g starts with $g(0) = r$ and ends with $g(1) = R$. It may also use a linear, convex, or concave interpolation along y. Its definition is

$$g(y) = \begin{cases} (R - r)(1 - (1 - y)^b) + r, & \textbf{if } T_y = concave \\ (R - r)y^b + r, & \textbf{if } T_y = convex \end{cases} \tag{2}$$

If b equals one, then both cases describe a linear gradient along y, which we also denote as $T_y = linear$. For $b \geq 2$, we get a concave shape with a decreasing gradient in the case $T_y = concave$ and a convex shape with increasing gradient for $T_y = convex$.

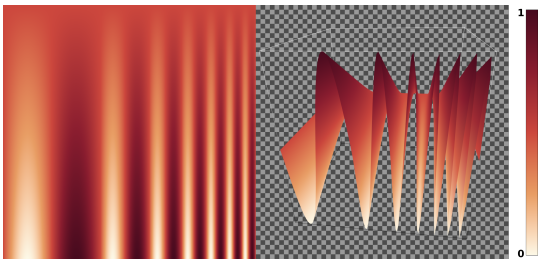

Fig. 6. These two pictures show an example for the `Frequency Variation` test. Left: 2D color mapping of the test field. Right: 3D height-map visualization.

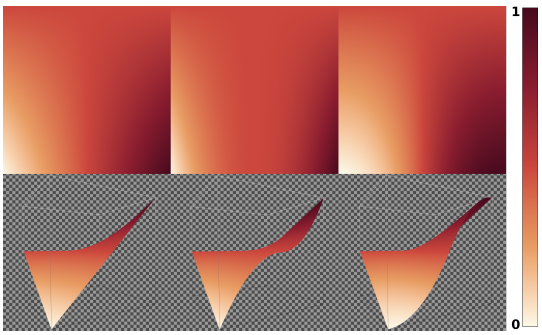

Fig. 7. This figure shows an example test image for each of the three `Threshold Variation` test types . From left to right you can see the linear, the flat, and the steep type. The top row shows the color mapping images and in the bottom row the corresponding test images as height-maps.

The actual test function is now defined in a similar manner by

$$f_{Grad}(x,y) = \begin{cases} (g(y)-r)(1-(1-x)^b)+r, & \textbf{if } T_x = concave \\ (g(y)-r)x^b + r, & \textbf{if } T_x = convex \end{cases} \quad (3)$$

Again, $b = 1$ means a linear gradient along x in both cases, and we can write $T_x = linear$ in this case. $T_x = concave$ means a convex shape created by a decreasing gradient, while $T_x = concave$ results in an increasing gradient and a concave shape. With $R > r$, we get an increasing function and with $R < r$ a decreasing one. Fig. 3 shows 2D and 3D visualizations of examples for the function $f_grad(x,y)$ for the three cases: $T_y = linear$, $concave$, and $convex$.

4.1.3 Local Topology: Minimum, Maximum, and Saddle

Besides discontinuities and gradients, local extrema and saddles are the next types of structures in scalar fields, according to calculus. This has also been noted by Ware [60] and is the fundamental insight behind topological visualization methods [17]. As local extrema (of some significance or persistence) and saddles are of interest for most scalar field analysis tasks, we thus include them in our test suite. For the calculation of user-definable minima, maxima, and saddles, we are using Equation 4.

$$f_{MMS}(x,y) = ox^2 + py^2 + m \quad (4)$$

It is based on the well-known fact that stable critical points can be described by quadratic functions. We put the extrema or saddle at zero. The user can create a maximum with $o < 0 \wedge p < 0$, a minimum with $o > 0 \wedge p > 0$, and a saddle with $o > 0 \wedge p < 0 \vee o < 0 \wedge p > 0$. The starting value of the structure is given by $m \in \mathbb{R}$. Fig. 4 shows an example of this test function with visualizations of minima, maxima, and saddle points.

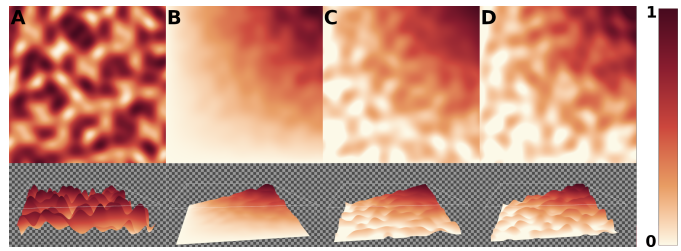

Fig. 8. This figure shows four color mappings of global topological structures with a gradient test function as foundation. The according 3D versions are shown at the bottom. The image **A** shows the pure noise using the *replacement* option. The following images show the options $max-scaled$ (**B**), $min-scaled$ (**C**) and $range-scaled$ (**D**) with activated clipping and a limited value adjustment of 0.25. While **B** focused the noise to higher foundation values, **C** does the same with lower ones. In contrast **D** allows globally adjustments of the foundation.

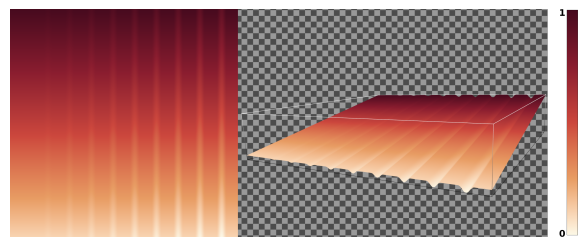

Fig. 9. The two pictures show the `Little Bit Variation` test ($m = 0.1$, $M = 1$, $g_m = 0.0001$, $g_M = 0.1$), which can used to test the impression of small value variations by the visibility of vertical grooves. The left image shows that the first groove with an amplitude of 0.0001 is not perceptible for all values between 0.1 and 1.0. The right image shows a 3D height-map visualization of the same test.

4.1.4 Local Topology: Ridge and Valley Lines

Besides local extrema and saddle points, ridge and valley lines are further relevant topological shape descriptors. Again, this has been noted by Ware [60] with respect to color mapping. Also, the relevance of ridges and valley lines is well established in feature-based flow visualization [17]. To test the suitability of colormaps for scalar fields that include such lines, we use a function $f_{RV} : [-1,1] \times [0,1] \to \mathbb{R}$. The location of the ridge/valley-line is always at $x = 0$ as a vertical line. Its shape is determined by the function g that we introduced in Sect. 4.1.2, so it may be linear, convex, or concave according to the exponent $b \in \mathbb{N}$ and the shape descriptor T_y. For the slope in x-direction, we basically use the absolute value with exponent b, i.e., $|x|^b$ on the interval $[-1,1]$. This creates a concave shape. For convex shapes, we use the similar function $1-(1-|x|)^b$. Both functions are adjusted to interpolate between r at -1 and 1 and $g(y)$ at 0. This is quite similar to the definition of g. We introduce the type parameter T_x and set it to "convex" or "concave" and arrive at the definition:

$$f_{RV}(x,y) = \begin{cases} (r-g(y))|x|^b + g(y) & \textbf{if } T_x = concave \\ (r-g(y))(1-(1-|x|)^b)+g(y) & \textbf{if } T_x = convex \end{cases} \quad (5)$$

As in the gradient variation case, $b = 1$ leads to the same linear function in both cases, which we also denote as "linear". For $R > r$, we get a ridge-line. For $R < r$, we get a valley-line. Fig. 5 shows an example test for ridge-lines using the different T_x and T_y types.

4.1.5 Frequency Variation

Another common test for analyzing colormaps involves variations of spatial frequencies [6, 29, 43]. To check the behavior for wave-like

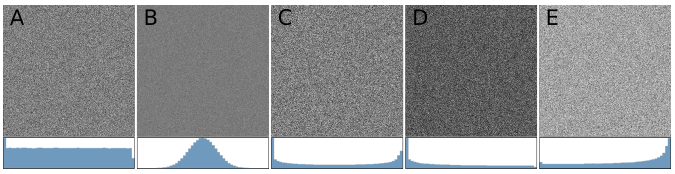

Fig. 10. These images show the different distributions for the signal noise option. To illustrate the noise, the pictures were created with the *replacement* option and a noise proportion of 100%. You can see the following distributions: **1**=uniform, **2**=normal, **3**=beta, **4**=beta-left, **5**=beta-right. Below each picture is a histogram of the distribution.

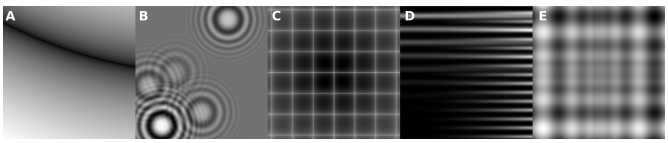

Fig. 11. This figure shows some examples for test functions used in the optimization computer science and also is of interest for our test function collection. From left to right you can see the functions: **A**:Bukin Function Number 6; **B**:Langermann Function; **C**:Cross-in-Tray Function; **D**:Levy Function Number 13; **E**:Schwefel Function. [18, 28]

functions, the `Frequency Variation Test` uses an increasing frequency in the x-direction and a decreasing amplitude in the y-direction. In x-direction, we use an additional parameter D defining the number of frequency increases. Basically, we start with a single sine wave with frequency 1. This takes one unit of space in x-direction. Then we add a single sine wave of frequency 2 using a half unit of space. Then, we add a single sine wave of frequency 3 using one third unit of space. We continue in this way until we have $D+1$ waves of increasing frequency. The x-range will obviously depend on the elements in the, so we denote their elements as

$$x_0 = 0 \quad x_j = \sum_{k=1}^{j} \frac{1}{k} \tag{6}$$

The wave will swing around the median value u with an amplitude W. Now, we can define the test function by

$$\begin{gathered} f_{Freq} : [x_0, x_{D+1}] \times [0,1] \to \mathbb{R} \\ f_{Freq}(x,y) = W(1-y)\sin(2\pi j(x-x_j)) + u, \quad x_{j-1} \le x \le x_j \\ j = 1, \ldots, D+1 \end{gathered} \tag{7}$$

Fig. 6 shows an example of the `Frequency Variation` test with six different frequencies. It should be noted that the image resolution is critical with regard to aliasing effects in case of high frequencies, i.e., when D is large.

4.1.6 Threshold Variation

In many scientific disciplines, natural thresholds, such as the freezing point of water at 0°C, and the data distribution close to them are of significant importance for visual analysis. This is also a major reason for the high relevance of isolines and isosurfaces in scientific visualization.

It is possible to integrate isolines into a colormap by creating a discontinuous transition point in the colormap [30]. With the `Threshold Variation` test, we created a function for testing a specific user-defined threshold t. The test function f_T spans the domain $[-1,1] \times [-1,1]$. The isoline with the value t is a vertical line in the middle, i.e., $x = 0$. On the left, the values are lower, and on the right, they are higher. We use two parameters, a minimum m, and a maximum M, with $m < t < M$. To increase visual information in the test image, the functions $f_M, f_m : [-1,1] \to \mathbb{R}$ change the minimum and maximum value linearly depending on y:

$$f_M(y) = \frac{M+t}{2} - \frac{M-t}{2} y \tag{8}$$

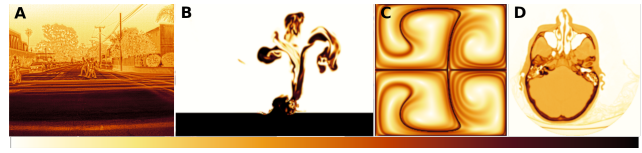

Fig. 12. This figure shows different real-world data sets we collected for our test suite. You can see the following data sets: **A**:A thermal data set for algorithm training (source: `https://www.flir.co.uk/oem/adas/adas-dataset-form/`); **B**:Simulation data of asteroid impacts in deep ocean water (source: `https://sciviscontest2018.org/`); **C**:FTLE technique from the field of flow visualization [48]; **D**:Medical computer tomography scan of a head (source: `https://graphics.stanford.edu/data/voldata/`).

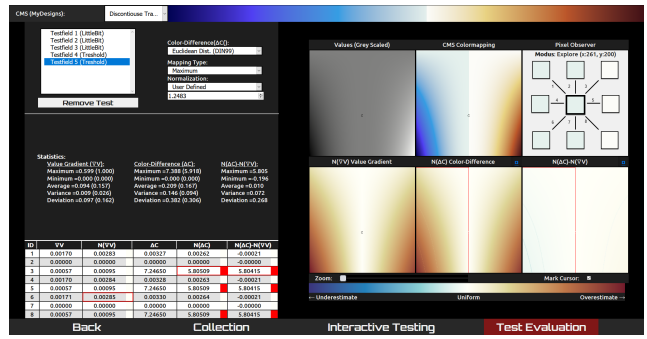

Fig. 13. This figure shows a screenshot of the test evaluation in the CCC-Tool. We present statistics of the `Value Difference Field`, `Color Difference Field`, and `Subtraction Field`. On the right side are two visualizations of the test-function. One color mapping is done with a grey-scaled colormap and the other one with the selected colormap for the analysis. Below are three color mapping images of the `Value Difference Field`, `Color Difference Field`, and `Subtraction Field`. All five images are zoomable and change interactively. As a sixth part, the pixel observer shows in combination with the table the pixel-neighborhood-information of the three fields.

$$f_m(y) = \frac{t+m}{2} + \frac{t-m}{2} y \tag{9}$$

We can now define our test function $f_T : [-1,1] \times [-1,1] \to \mathbb{R}$. Along each horizontal line, i.e., for fixed y, the value changes from $f_m(y)$ at -1 to t at 0, and finally reaches $f_M(y)$ at 1. We use three cases that we call $T = linear$, $T = flat$, and $T = steep$. The type describes the gradient behavior around the threshold. As in the cases for the gradient variation and ridge/valley lines, we use an exponent $b \in \mathbb{N}$ as final parameter. $b = 1$ describes the linear case, while higher exponents create a rather flat or steep gradient around the isoline. The formal definition of our test function is

$$f_T(x,y) = \begin{cases} (f_m(y)-t)|x|^b + t & \textbf{if } (T = flat \wedge x \le 0) \\ (f_M(y)-t)|x|^b + t & \textbf{if } (T = flat \wedge x > 0) \\ (f_m(y)-t)(1-(1-|x|)^b) + t & \textbf{if } (T = steep \wedge x \le 0) \\ (f_M(y)-t)(1-(1-|x|)^b) + t & \textbf{if } (T = steep \wedge x > 0) \end{cases} \tag{10}$$

In Fig. 7, a plot shows examples of all three types.

4.2 Global Tests

In contrast to the local tests, the global tests look at more global properties of scalar functions and how well the colormap presents them. First, we look at global topological properties. We use functions showing Perlin noise to create multiple local minima and maxima on different

height levels and different spatial structures. Details can be found in Sect. 4.2.1. Second, it is a challenge for any colormap to deal with a large span of the overall values while small, but relevant, local value variations are also present. Any nearly linear colormap will completely overlook these so-called little bit variations. As test functions, we use linear functions with varying gradient and height as background with small grooves to include little bit variations. The definition is given in Sect. 4.2.2. Third, real-world data, especially images created by measurements, contain noise of various types and intensity, i.e., signal-to-noise ratio. We use functions from the local test suite and add uniform or Gaussian distributed noise of different signal-to-noise ratios. We describe the details in Sect. 4.2.3. Finally, we add a collection of test functions from other computer science disciplines to allow for tests using these functions.

4.2.1 Global Topological Structures

As noted above, other authors indicated the relevance of critical points for testing colormaps before. In contrast to the local topology in Sect. 4.1.3, we use a larger number of critical points in the following test. For the creation of global topological structures, we take the 2D version of the improved noise algorithm introduced by Perlin [31, 32], which is often used for the creation of procedural textures or for terrain generation in computer games. The idea of this test function is to use some other test function from Sect. 4.1.2 to 4.2.2 as a background and combine this field with noise according to Perlin's work. These distorted gradients and shapes are in analogy with colormap testing functions specifically used to determine the discriminative power of subregions of colormaps [19, 29, 39, 61, 62].

To create the critical points, we use the noise function $f_{Noise}(x,y) \in [-n,n]$, $n > 0 \wedge n \leq 1$ and distinguish four options (see Fig. 8). For the options $min-$,$max-$, and $range-scaled$ the selected test-function f_{test} affect the result. Whereby the influence of the first two options depend on the closeness of the local value $f_{Noise}(x,y)$ to the m, or rather M. This procedure creates noise that is focused on small/high values. At the $range-scaled$ option, the adjustment of the local value is limited by the test-function range from m to M. Furthermore, an optional clipping method for these three options prevent values out of $[m,M]$. Fourth, we offer *replacement* as a final option, where users can set a custom noise-range $N = [n_m, n_M]$ with $f_{Noise}(x,y) \in N$. With this option, the entries of the test-function will be replaced by the noise value.

$$f_{test}(x,y) = \begin{cases} f_{test}(x,y) + f_{Noise}(x,y) * \frac{f_{test}(x,y)-m}{M-m} & \textbf{if } max-scaled \\ f_{test}(x,y) + f_{Noise}(x,y) * \frac{M-f_{test}(x,y)}{M-m} & \textbf{if } min-scaled \\ f_{test}(x,y) + f_{Noise}(x,y) * (M-m) & \textbf{if } range-scaled \\ f_{Noise}(x,y) & \textbf{if } replacement \end{cases} \tag{11}$$

4.2.2 Little Bit Variation

The teaser Fig. 1 demonstrates that standard colormaps may easily lead to overlooked small value variations. For such cases, i.e., if small variations in the scalar field (within a small sub-range of the full data range) carry valuable information for interpretation, we define a test on the potential of a given colormap to visually resolve small perturbations. This is similar to distorted gradients, which appear quite frequently in the literature [19, 29, 39, 61, 62]. The `Little Bit Variation` test

$$f_{LB} : [0, 2n+1] \times [0,1] \to \mathbb{R} \tag{12}$$

uses a background function and adds a function f_G producing n small grooves. The background function in this test is a linear gradient along the y-direction, which is defined by a user-specified value range $[m,M]$. Along the x-direction, this function is modified by a function f_G creating $2n+1$ alternate stripes of unchanged background and grooves, so we use

$$f_{LB}(x,y) = m + (M-m)y - f_G(x) \tag{13}$$

The function f_G produces sine-shaped grooves for odd $\lfloor x \rfloor$ and no changes for even $\lfloor x \rfloor$. As x runs from 0 to $2n+1$, this creates exactly n grooves.

$$f_G(x) = \begin{cases} 0, & \textbf{if } \lfloor x \rfloor \mod 2 = 0 \\ -f_A(x)\sin(\pi(x - \lfloor x \rfloor)), & \textbf{otherwise} \end{cases} \tag{14}$$

As can be seen, the sine wave's amplitude is changed by a function $f_A(x)$ and create a test of different small value changes (groove depths). The function $f_A(x)$ determine for each groove the depth by linear interpolation between user-defined minimum g_m and maximum g_M. In Fig. 9, you can see an example for the `Little Bit Variations` test.

$$f_A(x) = g_m + \frac{\lfloor x \rfloor - 1}{2*n-2}(g_M - g_m) \tag{15}$$

4.2.3 Signal-Noise Variation

In the signal and data processing, noise plays an important role. It also affects the results of scientific visualizations. Like the global topology test (see Sect. 4.2.1), our tool offers to add noise to each test function (Sect. 4.1.2 - Sect. 4.2.2). The tool uses the standard random algorithm from JavaScript, which produces pseud- random numbers in the range $[0,1]$ with uniform distribution. For the noise behavior, we offer the same noise behavior options from Sect. 4.2.1. Independent from the selected option, the fraction of noisy pixels can be set. This fraction describes how many randomly selected field values are affected by noise. If the noise proportion is set to 100%, the full test-function is affected by noise. For more flexibility, we also offer a conversion from a uniform distribution to a normal or a beta distribution. The conversion from uniform to the normal distribution is done with the Box-Muller transform [7]. With the normal distribution, the noise will be more focused on weaker changes around null for the $min/max-scaled$ and $range-scaled$ options. For the $replacment$ option, the normal distribution causes a focus on values around the median of the defined range of noise values. The approach from uniform to a beta-like distribution (with $alpha, beta = 0.5$) is done with the equation $beta_{Random} = sin(r * \frac{\pi}{2})^2$, with r being the result of the standard random generator. Adding noise using a beta distribution with the $min/max-scaled$ or $range-scaled$ options will have a priority for values near the maximal change parameter m and $-m$. For the *replacement* option, values near the minimum and maximum of the defined noise value range will be preferred. We modified this conversion with a view to do this preference on only one side, thus for m or $-m$ in the first case or for the maximum or minimum in the other case. The modification is a mirror at the median random value to the left or right side of this median. This allows us to create a beta-like distribution and also a left-oriented beta distribution and a right-oriented beta distribution. Fig. 10 shows the different distribution options.

4.2.4 Function Collection

Many domains of computer science use test functions for the evaluation of algorithms. There are several widespread well-known functions like `Mandelbrot Set` or `Marschner Lobb` and also functions like the `Six-Hump Camel Function` from the teaser, which are better known in optimization than computer science [18, 24, 25]. Such functions and their different attributes could also be an enrichment for evaluation in scientific visualization. Therefore, we included a collection of such functions from the literature in our testing environment. These functions stand beside our development of test functions, and provide further challenges for colormaps. With this collection, we want to provide over time more and more such functions of interest. In order to allow users to test their colormaps without changes, we allow the user to scale the values of these functions to the range of the colormap or a user-defined range. Fig. 11 shows some examples of functions used for optimization. Obviously, they also have relevant properties for the evaluation of color mapping. For example, the `Bukin Function` includes many small local minima along a valley-line. [18]

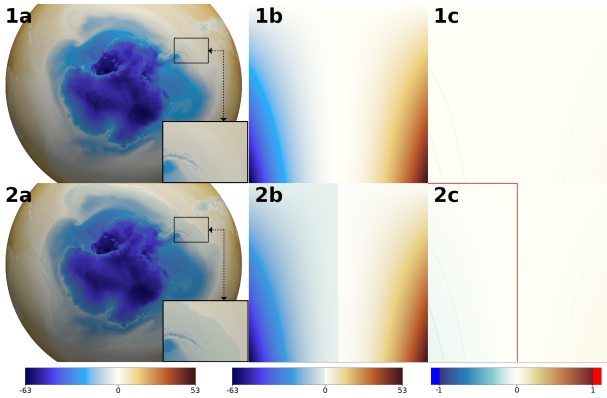

Fig. 14. This figure shows application of the *Threshold* test (Sect. 4.1.6) to improve the distinguishability of temperature variations around the South Pole with focus on the freezing point (see close-ups in the lower right corner of **a**). **1**: Local uniform optimized cool-warm colormap. **2**: Modified colormap with a discontinuous transition point to improve the threshold. **a**: Visualization of the 2m-temperature of a high resolution simulation with the global atmosphere model ICON. **b**: Threshold test function with $m = -63$, $M = 53$, and $t = 0$. **c**: `Subtraction Field` of the evaluation method (Sect. 5).

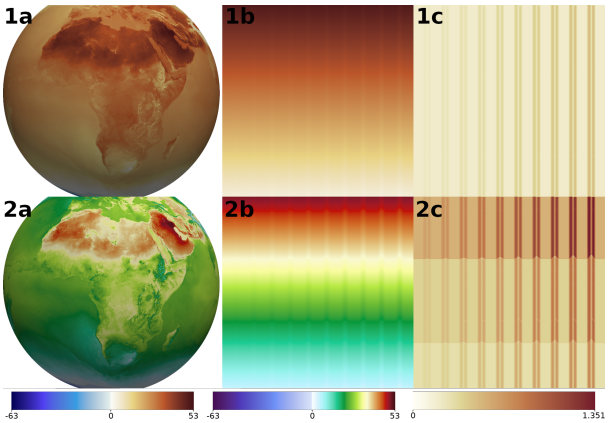

Fig. 15. **1**: Starting with the modified colormap of Fig. 14, we used the `Little Bit` test (Sect. 4.2.2) to increase the number of noticeable values of positives temperatures. The view in the **a**-panels is centered on Africa. **2**: Modified colormap which uses more hue variations to improve the `Little Bit` results. **a**: Visualization of the $2m$-temperature of a high resolution simulation with the global atmosphere model ICON. **b**: `Little Bit` test function with $m = 5°$C, $M = 53°$C, $g_m = 0.1°$C, and $g_M = 1.0°$C. **c**: `Color Difference Field` of the evaluation method (Sect. 5).

4.3 Application-Specific Tests

In the two previous sections, we described several analytic test functions concerning specific challenges encountered in color mapping. Additionally, we also introduced a collection of already existing test functions from other computer science domains. Nevertheless, we think that the involvement of real-world data is indispensable for the completion of this test suite. Real-world data originates from many different sources, is generated with various measurement techniques or simulation algorithms, and includes a myriad of attribute variations. Most importantly, such data could potentially present several of the challenges described in the two previous sections at the same time. This kind of test cannot be easily replaced by our theory-based test functions completely. Therefore, we decided to include a set of application test data from different domains to cover a wide spectrum of realistic challenges.

Within one specific scientific domain, there is often a similarity between typical data sets; e.g., in medicine, data from the MRI (Magnetic Resonance Imaging) or the CT (Computer Tomography) is frequently used. Such data sets have similar attributes, and similar requirements have to be fulfilled by colormaps. If we cover different typical data sets of different scientific disciplines, in the future, we can hopefully offer enough different real-world test cases so that most users will find a case that has some similarities with his data. Like the test function collection from Sect. 4.2.4, this collection of real-world data will be extended over time. At the current version, the tool offers medical-, flow-, and photograph-specific real-world data.

5 Test Evaluation

Mostly there are good reasons to select specific colormaps or to design colormaps in a specific way. Depending on the actually envisaged purpose of the colormap, a user decides on the number of keys; the hue, saturation and value of each key; the gradients in the mapping between the data range and the colormap; and so on. Furthermore, *de facto* standards and cognitive motivation may also influence the user's choice. Therefore, meaningful automated evaluation of continuous colormaps without knowledge of their intended use is rarely feasible. Therefore, a general colormap score computed based on automatic tests and benchmarks might not be informative.

Instead, we propose to derive information based on aforementioned test functions that can be analyzed and rated by users themselves. A user first chooses a test-function from Sect. 4. For each grid point of the generated test field, we calculate the value differences to the neighboring grid points. Depending on the location within the field, the number of neighbors varies between three and eight. We normalize these value differences with the minimum and maximum value differences found and save them into a `Value Difference Field`. We repeat this process also for the colors. Here, we use some color difference norm (Lab, DIN99, DE94, or CIEDE2000) and save the normalized values into the `Color Difference Field`.

By subtracting these two fields from another, we get a `Subtraction Field`. This field represents the local uniformity of the color mapping; when the local gradients found in the data are accordingly represented in the color mapped field, the difference between normalized data field and normalized color mapped field is zero for all pixels/locations. In the case of a non-linear color mapping, in contrast, the `Subtraction Field` will particularly highlight areas with strong non-linear mapping, which the user might have designed intentionally in order to increase the number of discriminable colors for a part of the data range. The user can study the `Color Difference Field` as well as the `Subtraction Field` to analyze the color mapping of the test function.

Each of the three fields has three up to eight values for each pixel. For the color mapping (Fig. 13), the user can select maximum, average, or median. Next to that, there are options to select a method for the calculation of the color difference. The tool offers Euclidean distance for Lab and DIN99 space or the use of the DE94 or CIEDE2000 metrics in the Lab space.

To compare the visualizations of `Color Difference Field` of different colormaps, we cannot use the normalization by minimum and maximum. The colors of such color mappings would relate to different color difference values and are not comparable. Therefore we implemented two alternative options using fixed values for the minimum and maximum of the normalization to create comparable results. The `Black-White` normalization use the greatest possible color difference between black and white as maximum and zero as minimum. The `Custom` normalization uses a user-entered maximum, which is a necessity if the black-white difference is to big by contrast with the occurring color differences of the `Color Difference Fields`. In Fig. 14, we used this third option to get a comparable visualization for a colormap with a discontinuous transition point.

6 Application Case

In this section, we show how the test suite could be utilized to evaluate the suitability of colormaps with respect to a given application problem. For this example, we chose a data set from a simulation with a high-resolution global atmosphere model. The data we use is one timestep

of the temperature at a height of 2m simulated with the icosahedral ICON model at a global resolution of 5km [53]. We remapped the data from the unstructured model grid to a regular grid with 4000×2000 grid points for easier use with different tools.

On the global scale, the 2m-temperature is typically characterized by a wide range of values between less than $-80°$C and more than $50°$C. For the selected time step, the simulated 2m-temperature varies between about $-63°$C and $52°$C. Regionally, however, small temperature variations of the order of $0.1°$C might be critical for the analysis as, e.g., in the neighborhood of the freezing point at $0°$C.

Panel **1a** of Fig. 14 shows a visualization of the data using a spherical projection with a focus on the South Pole. In contrast to mountainous regions, where the horizontal $2m$-temperature gradient is generally high, the gradient in flat areas such as oceanic regions is much smaller. Here, the color differences are too small to depict local temperature variations, as for example, in regions with values close to $0°$C as shown in the close-up in the lower right corner of the image.

To test a given colormap for its discriminative power in the data range around the freezing point, we applied a threshold test with the options $Flat - Surrounding$, $m = -63$, $M = 53$, and $t = 0$. First, we start with a local uniform cool-warm colormap (**1a** of Fig. 14). The related test function visualization **1b** demonstrates that it is impossible to differentiate between negative and positive values if the values are close to $0°$C. The **1c** `Subtraction Field` method of the test evaluation part (Sect. 5) yields a nearly white image, which reflects that the colormap uniformly represents the gradients produced by the test function. To highlight the freezing point in the mapping, we introduce a non-linearity in the colormap, at $0°$C. We use the twin key option of the CCC-Tool colormap specification (CMS) [30], which separates the color key at $0°$C into a left and right color key to create the discontinuous transition. To improve the visual difference between both sides, we slightly lower the lightness value and increase the left color saturation to achieve light blue. We kept white as the right-hand part of the color key. Panels **2a** and **2b** of Fig. 14 illustrates that the introduced discontinuity in the colormap clearly separates the areas with negative and positive temperature values. In comparison to **1c**, the `Subtraction Field` in **2c** shows with a vertical red line the spatial position of the discontinuous transition at $0°$C. The according visualization of the temperature field of the modified colormap is shown in panel **2a**.

If we visualize the global 2m-temperature field using a linear colormap and look at the tropics or the mid-latitudes, we find that regional variations are also not very well resolved. Using the same colormap, Fig. 15 **1a** shows a different view onto our planet, as Fig. 14 **1a**. The resolving power of the linear colormap is equally distributed over the full data range. However, when we analyze the global temperature distribution, we find that more than half of the data range is used for the temperature variations far below $0°$C mostly in Antarctica, although this information is less important for most users of such a data set. With respect to vegetation and agriculture, we may want to put more focus on regions with temperatures mostly above $0°$C.

Therefore we extended the path of the colormap through the color space to get more distinguishable colors for the positive data range. We used a `Little Bit` test to control improvements during this process. Panel **1a** of Fig. 15 shows a visualization using the colormap with the discontinuous transition introduced above. The corresponding `Little Bit` test is shown in panel **1b**. For the evaluation, we used the `Color Difference Field` (Sect. 5). Panel **1c** shows how the small grooves in the linear gradient of the `Little Bit` test function (that are hardly noticeable in **1b**) become clearly visible in the color difference field. From left to right, the regularly spaced perturbations in the field increase in magnitude, which is represented by a stripe pattern in panel **1c** that increases in contrast from left to right. The vertically constant color of the stripe pattern is a direct consequence of a linear colormap.

However, as we wanted to increase the discriminative power in the upper part of the colormap, we inserted additional color keys. First, we moved the blue part of the colormap representing negative values slightly away from cyan. The freed color space was utilized to represent the lower positive temperatures. A gradient from white to cyan $0°$C-$10°$C is followed by a gradient from cyan to green to represent the moderate temperature range of $0°$C-$20°$C. Next to this, a subsequent gradient from yellow through beige to light brown shows values between $20°$C and $40°$C. A further transition to dark red finally shows higher temperature range of up to $53°$C.

Our colormap semantics were designed to roughly differentiate between five temperature zones: very cold (blue to light blue), moderately cool (white to cyan), moderately warm (cyan to green), warm (green to yellow to beige) and hot (red). Concerning red-green colorblind viewers, we used a lower and not overlapping lightness range for the red gradient and the green gradient. The respective color gradients were separately optimized for local uniformity. The panels **2a** and **2b** of Fig. 15 show the visualizations of the temperature data and the test function with the modified colormap. Note that we used the `Little Bit` test function only for the upper part of the colormap that corresponds to temperature values between $m = 5°$C and $M = 53°$C. As a result of our modifications of the colormap, it is now possible to see much more detail in the inhabited part of our planet and to distinguish between the different temperature zones. Compared to **1c**, the `Color Difference Field` **2c** shows an increase in the color difference at the expense of the local uniformity of the positive data range.

7 Conclusion

In this paper, we have introduced the approach of using test functions as a standard evaluation method, and we have presented a test suite for continuous colormaps. Like in other fields of computer science, one could use such test functions besides user-centered evaluation (e.g., user testimonies and empirical studies). In compassion with user-centered evaluation, there is no need to recruit participants, design questionnaires or stimuli, organize payment, arrange experiment time and environment, and provide apparatus. Evaluating colormaps using the test suite can be conducted quickly and easily. The designer can test many optional colormaps against many test functions and data sets, which is usually not feasible with user-centered evaluation. The same tests can be repeated with consistent control and comparability.

For the test suite, we first focused on the specific challenges of scalar fields. The Sect. 4.1.1-4.2.2 describe the test functions we chose to address these challenges. To help users with a less mathematical background, we tried to develop intuitive functions that are simple and easy to interpret. The test suite currently includes step functions, different gradients, minima, maxima, saddle points, ridge and valley lines, global topology, thresholds, different frequencies, and a test for very small value changes. Although these test functions cannot cover all possible challenges, we have laid down a solid foundation that can be extended continually. We have also included the option to add noise to extend the possibilities of the basic test functions.

Besides our newly designed functions, we have presented in Sect. 4.2.4 a collection of functions used for evaluation in other computer science fields. We think they will prove to be useful for the evaluation of colormaps as well. Furthermore, we have included an initial selection of real-world data sets from different application areas. As described in Sect. 4.3, tests against real-world data are important in practice. Each real-world data set in our test suite presents an individual challenge of a combination of challenges in scalar field analysis. Here, our intention is to provide a broad cover such that users are less dependent on external data.

Our test suite has been integrated into the open-access CCC-Tool. In Sect. 5 we describe means to evaluate the results of the test functions visually and numerically that we have also implemented into our online-tool. An example of using the test suite to evaluate and enhance a user-designed colormap concerning a specific application problem is finally presented and discussed in Sect. 6.

For a long-term perspective, we plan to continue the extension of our collection. One option for real world data would be an open source database with a web interface and a link to our tool. In order to adopt the test suite as a standard evaluation method, we would like to work on the method of automatic test reports, which can perform automatic analysis of a colormap with a set of tests chosen by the user.

REFERENCES

[1] H. E. Aguirre and K. Takadama, eds. *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018, Kyoto, Japan, July 15-19, 2018*. ACM, 2018. doi: 10.1145/3205455

[2] J. Ahrens, B. Geveci, and C. Law. 36 - paraview: An end-user tool for large-data visualization. In C. D. Hansen and C. R. Johnson, eds., *Visualization Handbook*, pp. 717 – 731. Butterworth-Heinemann, Burlington, 2005. doi: 10.1016/B978-012387582-2/50038-1

[3] 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*, p. 118. IEEE Computer Society, 1995.

[4] J. Bernard, M. Steiger, S. Mittelstädt, S. Thum, D. Keim, and J. Kohlhammer. A survey and task-based quality assessment of static 2D colormaps. In *SPIE/IS&T Electronic Imaging*, pp. 93970M–93970M. International Society for Optics and Photonics, 2015.

[5] D. Borland and A. Huber. Collaboration-specific color-map design. *IEEE Computer Graphics and Applications*, 31(4):7–11, July 2011. doi: 10.1109/MCG.2011.55

[6] D. Borland and R. M. Taylor II. Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 27(2):14–17, 2007.

[7] G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, 1958.

[8] C. Brewer. *Designing Better Maps: A Guide for Gis Users*. Environmental Systems Research, 2004.

[9] C. A. Brewer. Color use guidelines for mapping. *Visualization in modern cartography*, pp. 123–148, 1994.

[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*, 24(1):923–933, Jan 2018. doi: 10.1109/TVCG.2017.2743978

[11] R. E. Burkard, S. E. Karisch, and F. Rendl. Qaplib – a quadratic assignment problem library. *Journal of Global optimization*, 10(4):391–403, 1997.

[12] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rubel, M. Durant, J. Favre, and P. Navratil. Visit: An end-user tool for visualizing and analyzing very large data. *High Performance Visualization-Enabling Extreme-Scale Scientific Insight*, pp. 357–372, 10 2012.

[13] M. Eisemann, G. Albuquerque, and M. Magnor. Data driven color mapping. In *Proceedings of EuroVA: International Workshop on Visual Analytics, Bergen, Norway*. Citeseer, 2011.

[14] 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*, 23(1):871–880, Jan 2017. doi: 10.1109/TVCG.2016.2599214

[15] C. Floudas and P. Pardalos. *Handbook of test problems in local and global optimization*. Nonconvex optimization and its applications. Kluwer Academic Publishers, 1999.

[16] D. L. Gresh. Self-corrected perceptual colormaps. Technical report, IBM, 2008. RC24542 (W0804-104).

[17] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. A survey of topology-based methods in visualization. *Computer Graphics Forum*, 35(3):643–667, 2016.

[18] M. Jamil and X.-S. Yang. A literature survey of benchmark functions for global optimization problems. *CoRR*, abs/1308.4008, 2013.

[19] A. D. Kalvin, B. E. Rogowitz, A. Pelah, and A. Cohen. Building perceptual color maps for visualizing interval data. In *Human Vision and Electronic Imaging V*, vol. 3959, pp. 323–336. International Society for Optics and Photonics, 2000.

[20] G. Kindlmann, E. Reinhard, and S. Creem. Face-based luminance matching for perceptual colormap generation. In *Proceedings of the conference on Visualization'02*, pp. 299–306. IEEE Computer Society, 2002.

[21] H. Levkowitz and G. T. Herman. The design and evaluation of color scales for image data. *IEEE Computer Graphics and Applications*, 12(1):72–80, 1992.

[22] A. Light and P. J. Bartlein. The end of the rainbow? Color schemes for improved data graphics. *Eos*, 85(40):385–391, 2004.

[23] M. R. Luo, G. Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application*, 26(5):340–350, 2001.

[24] B. B. Mandelbrot. Fractal aspects of the iteration of z z(1- z) for complex and z. *Annals of the New York Academy of Sciences*, 357(1):249–259, 1980. doi: 10.1111/j.1749-6632.1980.tb29690.x

[25] S. R. Marschner and R. J. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proceedings Visualization '94*, pp. 100–107, 1994.

[26] S. Mittelstädt, D. Jäckle, F. Stoffel, and D. A. Keim. ColorCAT: Guided Design of Colormaps for Combined Analysis Tasks. In E. Bertini, J. Kennedy, and E. Puppo, eds., *Eurographics Conference on Visualization (EuroVis) - Short Papers*, pp. 115–119. The Eurographics Association, 2015.

[27] S. Mittelstädt, A. Stoffel, and D. A. Keim. Methods for compensating contrast effects in information visualization. In *Computer Graphics Forum*, vol. 33, pp. 231–240. Wiley Online Library, 2014.

[28] M. Molga and C. Smutnicki. Test functions for optimization needs. 2005.

[29] K. Moreland. Diverging color maps for scientific visualization. In *International Symposium on Visual Computing*, pp. 92–103. Springer, 2009.

[30] P. Nardini, M. Chen, F. Samsel, R. Bujack, M. Böttinger, and G. Scheuermann. The making of continuous colormaps. *IEEE Transactions on Visualization and Computer Graphics*, 2019. doi:10.1109/TVCG.2019.2961674.

[31] K. Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, July 1985. doi: 10.1145/325165.325247

[32] K. Perlin. Improving noise. *ACM Trans. Graph.*, 21(3):681–682, July 2002. doi: 10.1145/566654.566636

[33] S. H. Pizer, J. B. Zimmerman, and R. E. Johnston. Concepts of the display of medical images. *IEEE Transactions on Nuclear Science*, 29(4):1322–1330, 1982.

[34] S. M. Pizer. Intensity mappings to linearize display devices. *Computer Graphics and Image Processing*, 17(3):262–268, 1981.

[35] G. Reinelt. Tsplib – a traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.

[36] P. L. Rheingans. Task-based color scale design. In *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*, pp. 35–43. International Society for Optics and Photonics, 2000.

[37] P. K. Robertson and J. F. O'Callaghan. The generation of color sequences for univariate and bivariate mapping. *IEEE Computer Graphics and Applications*, 6(2):24–32, 1986.

[38] 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*, pp. 183–556. IEEE, 2001.

[39] B. E. Rogowitz, A. D. Kalvin, A. Pelah, and A. Cohen. Which trajectories through which perceptually uniform color spaces produce appropriate colors scales for interval data? In *7th Color and Imaging Conference*, pp. 321–326. Society for Imaging Science and Technology, 1999.

[40] B. E. Rogowitz, D. T. Ling, and W. A. Kellogg. Task dependence, veridicality, and preattentive vision: taking advantage of perceptually rich computer environments. In *Human Vision, Visual Processing, and Digital Display III*, vol. 1666, pp. 504–514. International Society for Optics and Photonics, 1992.

[41] B. E. Rogowitz and L. A. Treinish. Using perceptual rules in interactive visualization. In *Human Vision, Visual Processing, and Digital Display V*, vol. 2179, pp. 287–296. International Society for Optics and Photonics, 1994.

[42] B. E. Rogowitz and L. A. Treinish. Data visualization: the end of the rainbow. *IEEE spectrum*, 35(12):52–59, 1998.

[43] B. E. Rogowitz, L. A. Treinish, and S. Bryson. How not to lie with visualization. *Computers in Physics*, 10(3):268–273, 1996.

[44] 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)*, pp. 703–710, 2015.

[45] F. Samsel, M. Petersen, T. Geld, G. Abram, J. Wendelberger, and J. Ahrens. Colormaps that improve perception of high-resolution ocean data. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pp. 703–710, 2015. doi: 10.1145/2702613.2702975

[46] F. Samsel, T. L. Turton, P. Wolfram, and R. Bujack. Intuitive Colormaps for Environmental Visualization. In R. Bujack, A. Middel, K. Rink, and D. Zeckzer, eds., *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, pp. 55–59. The Eurographics Association, 2017. doi: 10.2312/envirvis.20171105

[47] P. Schulze-wollgast, C. Tominski, and H. Schumann. Enhancing visual exploration by appropriate color coding. In *Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG*, pp. 203–210, 2005.

[48] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties

of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3-4):271–304, 2005.

[49] 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, 2007. IV'07. 11th International Conference*, pp. 943–950. IEEE, 2007.

[50] S. Silva, B. S. Santos, and J. Madeira. Using color in visualization: A survey. *Computers & Graphics*, 35(2):320–333, 2011.

[51] K. R. Sloan and C. M. Brown. Color map techniques. *Computer Graphics and Image Processing*, 10(4):297–317, 1979.

[52] D. Stalling, M. Westerhoff, H.-C. Hege, et al. Amira: A highly interactive system for visual data analysis. *The visualization handbook*, 38:749–67, 2005.

[53] B. Stevens, C. Acqistapace, A. Hansen, R. Heinze, C. Klinger, D. Klocke, H. Rybka, W. Schubotz, J. Windmiller, P. Adamidis, I. Arka, V. Barlakas, J. Biercamp, M. Brueck, S. Brune, S. Buehler, U. Burkhardt, G. Cioni, M. Costa-Suros, S. Crewell, T. Crger, H. Deneke, P. Friederichs, C. Henken, C. Hohenegger, M. Jacob, F. Jakub, N. Kalthoff, M. Köhler, T. van Laar, P. Li, U. Löhnert, A. Macke, M. Madenach, B. Mayer, C. Nam, A. Naumann, K. Peters, S. Poll, J. Quaas, N. Röber, N. Rochetin, L. Scheck, V. Schemann, S. Schnitt, A. Seifert, F. Senf, M. Shapkalijevski, C. Simmer, S. Singh, O. Sourdeval, D. Spickermann, J. Strandgren, O. Tessiot, N. Vercauteren, J. Vial, A. Voigt, and G. Zängl. The added value of large-eddy and storm-resolving models for simulating clouds and precipitation. *Journal of the Meteorological Society of Japan. Ser. II*, advpub, 2020. doi: 10.2151/jmsj.2020-021

[54] J. Tajima. Uniform color scale applications to computer graphics. *Computer Vision, Graphics, and Image Processing*, 21(3):305–325, 1983.

[55] Text REtrieval Conference (TREC). Data. `https://trec.nist.gov/data.html`, accessed in July 2020.

[56] D. C. Thompson, J. Bennett, C. Seshadhri, and A. Pinar. A provably-robust sampling method for generating colormaps of large data. In *LDAV*, pp. 77–84, 2013.

[57] C. Tominski, G. Fuchs, and H. Schumann. Task-driven color coding. In *2008 12th International Conference Information Visualisation*, pp. 373–380. IEEE, 2008.

[58] H. Wainer and C. M. Francolini. An empirical inquiry concerning human understanding of two-variable color maps. *The American Statistician*, 34(2):81–93, 1980.

[59] L. Wang, J. Giesen, K. McDonnell, P. Zolliker, and K. Mueller. Color design for illustrative visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1739–1754, Nov 2008. doi: 10.1109/TVCG.2008.118

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

[61] C. Ware, T. L. Turton, F. Samsel, R. Bujack, and D. H. Rogers. Evaluating the Perceptual Uniformity of Color Sequences for Feature Discrimination. In K. Lawonn, N. Smit, and D. Cunningham, eds., *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)*, pp. 7–11. The Eurographics Association, 2017. doi: 10.2312/eurorv3.20171107

[62] C. Ware, T. L. Turton, F. Samsel, R. Bujack, D. H. Rogers, K. Lawonn, N. Smit, and D. Cunningham. Evaluating the perceptual uniformity of color sequences for feature discrimination. In *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)*. The Eurographics Association, 2017.

[63] Wikipedia. Semeval (semantic evaluation). `https://en.wikipedia.org/wiki/SemEval`, accessed in July 2020.

[64] A. Zeileis, K. Hornik, and P. Murrell. Escaping RGBland: selecting colors for statistical graphics. *Computational Statistics & Data Analysis*, 53(9):3259–3270, 2009.

[65] H. Zhang and E. Montag. Perceptual color scales for univariate and bivariate data display. *The Society for Imaging Science and Technology (IS&T)*, 2006.

[66] L. Zhou and C. Hansen. A survey of colormaps in visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(8):2051–2069, 2016.