





# Automatic Improvement of Continuous Colormaps in Euclidean Colorspaces

Pascal Nardini<sup>1</sup> , Min Chen<sup>2</sup> , Michael Böttinger<sup>3</sup>, Gerik Scheuermann<sup>1</sup> , and Roxana Bujack<sup>4</sup> 

<sup>1</sup>Institute of Computer Science, University of Leipzig, Germany

<sup>2</sup>Department of Engineering Science, University of Oxford, UK

<sup>3</sup>German Climate Computing Center (DKRZ), Hamburg, Germany

<sup>4</sup>Data Science at Scale Team, Los Alamos National Laboratory, Los Alamos, NM, USA

## Abstract

*Colormapping is one of the simplest and most widely used data visualization methods within and outside the visualization community. Uniformity, order, discriminative power, and smoothness of continuous colormaps are the most important criteria for evaluating and potentially improving colormaps. We present a local and a global automatic optimization algorithm in Euclidean color spaces for each of these design rules in this work. As a foundation for our optimization algorithms, we used the CCC-Tool colormap specification (CMS); each algorithm has been implemented in this tool. In addition to synthetic examples that demonstrate each method's effect, we show the outcome of some of the methods applied to a typhoon simulation.*

## 1. Introduction

Colormapping in scientific visualization is one of the most common methods for scientists to gain insight into their data and communicate their research, and the colormap choice can have a large impact on the effectiveness of a visualization. A sub-optimal colormap may hide valuable information or introduce spurious artifacts. There are empirical design guidelines that have been suggested and motivated throughout the literature on colormaps. A rigorous collection [BTS\*18] showed that the most important ones are uniformity, discriminative power, order, and smoothness, but they are usually only applied in an intuitive fashion for three reasons.

First, they are partly contradictory: a reasonable trade-off needs to be found based on the visualization goal. Second, the theory is not sufficiently developed to use a purely algorithmic approach. Algorithms have been suggested to make a colormap locally uniform [RO86, Gre08] and smooth [Pha90], but to the best of our knowledge, there are no algorithms available from the literature that improve a colormap w.r.t. any of the other design rules. Third, there is no openly available tool for the application of any of these rules.

In this work, we present algorithms that are able to transform a colormap into one that performs more effectively w.r.t. a specific colormap design rule while distorting the original colormap as little as possible. Specifically, we propose algorithms for improving colormaps with respect to the following four design rules [BTS\*18].

**Uniformity** is the concept that perceived distances between colors are proportional to the distances of the values they represent. It is important to correctly represent the structure of underlying data;

a non-uniform colormap may hide features in the visualization or introduce artifacts.

**Order** is the concept that an observer should be able to order the colors the colormap traverses so that she/he can identify which values are higher or lower. For example, greyscaled colors can be placed in a consistent perceptual order, from darker to lighter.

**Discriminative power** can be thought of as the contrast in the visualization. It is described as the number of just noticeable differences (JNDs) traversed by the colormap [Piz81]. Greater discriminative power means greater ability to see details in the data.

**Smoothness** is the absence of sudden bends in the path of the colormap that cause artificial bands in the visualization.

We selected and developed the algorithms based on the following three criteria:

- How well each algorithm succeeds in improving a colormap w.r.t. the corresponding design rule;
- Their capacity to run in an interactive environment; and
- How much they distort the original colormap.

In a nutshell, the contributions of this paper are as follows:

- To the best of our knowledge, we provide the first algorithm for improving a colormap w.r.t. global uniformity as well as order and discriminative power.
- We provide implementations of all mentioned design rules in an open source online tool for colormap design, analysis, improvement, and testing at <https://ccctool.com/>.
- We demonstrate the performance in several demonstrative examples and a real world application.

## 2. Related Work

Colormapping has been a pervasive topic in the visualization literature. There are articles presenting critical analysis (e.g., [RT98]), offering recommendations (e.g., [HB03]), reporting empirical studies (e.g., [Sza18]), and proposing technical solutions (e.g., [NCS\*19]). As this work is concerned with a technical solution, we focus our literature review on design rules, tools, and algorithms for colormapping.

### 2.1. Design Rules for Colormaps

In recent years, several surveys collected a variety of design rules, recommendations, and best practices for the design of colormaps [SMS07, SSM11, ZH16, Kov15]. Bujack et al. [BTS\*18] analyzed these design rules and identified the most popular properties: *order*, *discriminative power*, *uniformity*, and *smoothness*.

In many situations, design rules may conflict with each other, and finding a colormap that performs well with respect to all applications is difficult [Mor09]. A broad body of work discusses compromises that have to be made with respect to the domain-specific task, the shape of the data, the audience, the display, or the visualization goal [War88, BRT95, RTB96, Rhe00, TFS08].

Even well-defined design rules are not straightforward how they can aid the automatic generation, evaluation, and improvement of colormaps. One needs a consistent measurement for quantifying the quality of colormaps with respect to a given design rule. A number of suggestions for quantitative measurements have been made [Taj83, RO86, LH92, Mor09, BTS\*18], and they are mostly based on uniform colorspaces [oi04, LCL06, MEO94, HMO06]. Alternatively, measured data can be obtained using empirical studies, which implicates the effectiveness and helps select colormaps preferred by some observers, appropriate to some tasks, or optimal for some monitors [War88, RKPC99, KRPC00, Gre08, WTS\*17].

In this work, we build on the measurement proposed by Bujack et al. [BTS\*18] and develop algorithms for improving colormaps.

### 2.2. Tools for Colormapping

Many visualization tools (e.g., [AGL05, CBW\*12]) provide default colormaps and colormap editors. Some colormap creation tools support the process of selecting a recommended colormap or defining a new colormap. ColorBrewer [HB11] provides a collection of carefully designed discrete colormaps. PRAVDAColor [BRT95] uses a set of pre-defined rules to recommend colormaps according to visualization tasks, data types, and spatial frequencies. ColorCAT [MJSK15a] provides users with guidance on ordering, color vision deficiencies, and perceived color differences. iWantHue creates discrete colormaps based on custom specifications of hue, chroma, and luminance [Jac13]. Other tools such as Color Hex, Adobe Color CC, and COLRD enable collective decisions by allowing users to generate, share, rate, and/or comment on colormaps. ColorCrafter [SWS19] allows a user to specify a seed color, and generates a sequential colormap with the aid of a corpus of expert-designed color ramps. VisCM [Hun07], allows users to create near-uniform colormaps with linearly increase of luminance. ColorMoves [SKR18] provides users with a large set of pre-defined

colormap segments and allows them to define customized colormaps aligning with their specific data in order to facilitate feature identification and exploration. ColorMeasures [BTS\*18] analyzes the quality of a colormap in relation to several popular design rules, informing users of characteristics such as discriminatory power, uniformity, and the other properties. CCC-Tool provides users with several aforementioned functions, such as offering colormaps, creating and editing continuous, discrete, and combined colormaps, specifying customized colormaps with application-specific semantics, and analyzing the colormap properties.

However, none of these tools is equipped with algorithms for automatically improving a colormap defined by users based on their application needs. In this work, we present a major extension of *CCC-Tool* [NCS\*19], introducing new algorithms developed based on the theoretical framework outlined in [BTS\*18]. With the existing facilities of *CCC-Tool* and its web-based interface for cross-platform availability, these algorithms empower the user to make informed decisions about where to follow the design rules closely and where to relax them for the benefit of application-specific semantics, tasks, users, and visualization goals.

### 2.3. Algorithms for Colormap Improvement

A number of algorithms have been proposed to improve some properties defined by design rules. Several algorithms focus on **discriminative power**. The algorithm by Levkowitz and Herman [LH92] maximizes color differences while maintaining monotonicity in RGB, hue, saturation, and brightness. The algorithm by Fang et al. [FWD\*17] is designed for improving discrete colormaps, and it moves colors in a perceptually-uniform colorspace with the user-definable constraints of hue, saturation, and brightness. They compared three different optimization methods for controlling the movement of the colors.

Some algorithms focus on **uniformity**. The algorithm by Robertson and O'Callaghan [RO86] converts a univariate colormap to a perceptually-uniform colorspace, and redistributes the points in the colormap to ensure equal distances between adjacent points along the curve in the perceptually-uniform colorspace. In addition, they propose three algorithms for improving the uniformity of bivariate colormaps. The algorithm by Levkowitz first supersamples a given colormap through linear interpolation and then selects a subset of the points that have approximately equal distances as the new control points. The algorithm is more efficient, as the process of "linearization and equalization" is a local optimization process. The algorithm by Gresh [Gre08] uses measured perceptual differences related to a colormap, a user, and a monitor and normalizes the colormap to achieve "equal perceptual steps".

In addition, for **smoothness**, the algorithm by Pham [Pha90] produces *low curvature* colormaps by fitting splines through the given key points in a colorspace.

While the above algorithms improve colormaps independent of individual datasets, there are also algorithms that modify a colormap for each dataset in order to maximize some properties. For such **data-driven improvement**, the algorithm by Schulze-Wollgast et al. [SwTS05] automatically calculates a set of statistical measures of the data, associates key colors with these values,

and generates intermediate points in the colormap. The algorithms by Tominski et al. [TFS08] optimize a colormap based on data in order to support a visualization task better. For example, one algorithm generates a “histogram-equalized” colormap that maximizes the discrimination of visual objects in a particular dataset. The algorithm by Eisemann et al. [EAM11] also makes use of the histogram of the data to maximize the discrimination of color dots in choropleth maps. The algorithm by Zeng et al. [ZWZ\*19] uses an optimization method based on energy minimization to “deform” a colormap non-linearly to improve the discrimination of some sampled points in a scalar field with their neighbors.

CCC-Tool was designed to address the needs for creating colormaps that encode semantics in an application. On one hand, such a colormap is independent from an individual dataset, facilitating consistent comparison among visualization images created for different datasets. On the other hand, such a colormap encodes some characteristic information about all or most possible datasets in an application context, such as critical values, important ranges, etc.. Ensuring a design rule universally across the whole colormap may remove such characteristic information. Therefore, it is desirable to develop a set of algorithms that can be deployed in CCC-Tool without eroding application-specific semantics. This work focuses on such algorithms.

### 3. Foundations

The following definitions are taken from Bujack et al. [BTS\*18].

**Definition 1** A colormap is a function  $x : [a, b] \subset \mathbb{R} \rightarrow C$ , which is defined by a color space  $C$ , an increasing sequence of sampling points  $t_0 < \dots < t_m \in [a, b]$ , a series of values in the color space  $x_0, \dots, x_m \in C$ , the mapping  $x(t_i) = x_i, i = 0, \dots, m$ , and a rule for interpolating the intermediate values  $t_{i-1} < t < t_i \in [a, b]$ .

If not explicitly stated otherwise, we will simplify the notation by assuming that our colormap is given on the unit interval in a Euclidean color space using linear interpolation between sufficiently dense, equidistantly sampled points  $\forall i \in \{0, \dots, n\} : t_i = \frac{i}{n} \in [0, 1]$  for the remainder of this paper.

For a colormap, the mutual distances between all pairs of points  $t_i, t_j, i, j \in \{0, \dots, n\}$  give a 2D matrix  $D \in \mathbb{R}^{(n+1) \times (n+1)}$ :

$$D_{i,j} := \Delta E_{ij} = \Delta E(x(t_i), x(t_j)). \quad (1)$$

Additionally, the distance allows for a definition of the metric speed  $\forall i \neq j \in \{0, \dots, n\} : V \in \mathbb{R}^{(n+1) \times (n+1)}$ ,

$$V_{i,j} := \frac{D_{i,j}}{|t_j - t_i|}. \quad (2)$$

We can derive the local distance vector  $d : \in \mathbb{R}^n$  between two neighboring points on the colormap  $\forall j \in \{1, \dots, n\}$ :

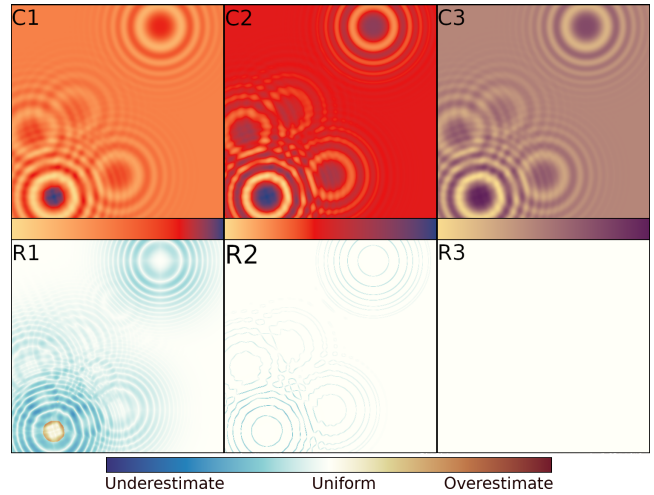
$$d_j := D_{j,j-1} \quad (3)$$

and analogously the local speed  $v_j \in \mathbb{R}^n$

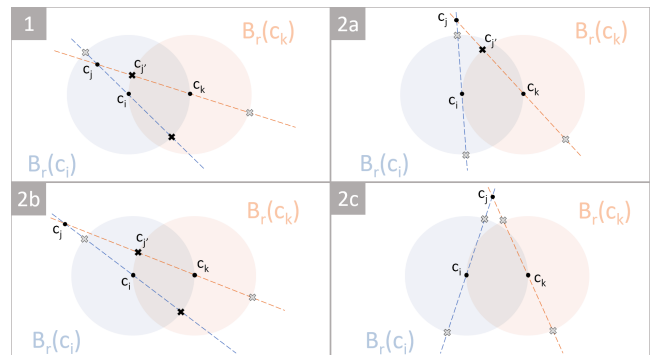
$$v_j := V_{j,j-1}. \quad (4)$$

In the coming sections, we will also use the average local speed  $\bar{v} \in \mathbb{R}$ :

$$\bar{v} := \frac{\sum_{j=1}^n v_j}{n} \quad (5)$$



**Figure 1:** Local and global uniformity applied to the Lab colorspace. **C-Row:** Colormapping visualization of the Langermann test-function [MS05]. **R-Row:** Test-function report of the CCC-Tool, details in Sec. 5



**Figure 2:** The four possible cases that can be encountered if the optimal position for the color  $c_j$  has to be determined to provide intrinsic order. The individual cases are explained in Section 4.2. Black crosses show intersections fulfilling intrinsic order and possible positions for  $c_j$ . Grey ones wouldn't provide intrinsic order.

and the average global speed  $\bar{V} \in \mathbb{R}$ :

$$\bar{V} := \frac{\sum_{i,j=0}^n V_{i,j}}{n(n+1)}. \quad (6)$$

Please note that in practice, the concepts of distance and speed and the related algorithms are only as meaningful as the metric in the used colorspace. Their values differ if evaluated in different colorspace just like the appearance of the colormap differs when interpolated in different spaces. Furthermore, there are visual effects, e.g., contrast effects [MSK14], ambient illumination [RFB16], feature size [SSS14], and spatial frequency [WTB\*18, RNAK18], that come from the final placement of the colors. They cannot be measured without taking the data into account and we will not treat them in this work.

Perceptual Rule	Optimization Criterion	Improvement Algorithm
4.1.1 Local Uniformity	minimize deviation of local speed	reparameterization by arc length
4.1.2 Global Uniformity	minimize deviation of global speed	linear regression
4.3.1 Local Legend-based Order	positive minimum local speed/distance	force based algorithm with pairs of neighboring points repelling each other if they are below the threshold of $1JND/(t_n - t_0)$
4.3.2 Global Legend-based Order	positive minimum global speed/distance	force based algorithm with pairs of points (including edges) repelling each other if they deceed the threshold of $1JND/(t_n - t_0)$
4.2 Local Intrinsic Order	positive local triangle side difference	relocation of violator to closest point in the triangle w.r.t. neighbors
4.2 Global Intrinsic Order	positive global triangle side difference	relocation of violator to closest point in the triangle w.r.t. all colors
4.4 Local Discriminative Power	maximize average local speed/distance	force based algorithm with pairs of neighboring points repelling each other if they deceed the local speed threshold of $ black - white /(t_n - t_0)$ times number of segments
4.4 Global Discriminative Power	maximize average global speed/distance	force based algorithm with pairs of points (including edges) repelling each other if they deceed the average global speed of the <i>z</i> -order curve
4.5.1 Local Intrinsic Smoothness	minimize deviation of local curvature	relocation of violator to closest point on the circle w.r.t. neighbors
4.5.1 Global Intrinsic Smoothness	minimize deviation of global curvature	relocation of violator to closest point on the circle w.r.t. all colors

**Table 1:** Summary of the treated perceptual rules, the targeted optimization, criterion and the algorithms implemented to achieve the corresponding improvement.

## 4. Algorithms

In this section, we introduce both a local and a global optimization algorithm for each of the different evaluation rules for Euclidean color spaces (see Table 1). The algorithms are developed for the optimization of the new colormap specification (CMS) used by the CCC-Tool [NCS\*19]. This specification is designed to reduce a colormap to a minimum number of key colors  $c$ .

In practical applications, the true optimum is usually far away from the input colormap and not at all what the user wants to obtain. We allow the users to select a subregion of their colormap for the optimization, which leaves the remaining parts unchanged. Further, in order to get customized results, we offer the user a slider to obtain an interpolated map between original CMS and the optimized CMS.

Applying the following optimization methods can cause a key color movement outside of the selected interpolation color space. In such a case, it is necessary to apply a *Clipping* method. As the CCC-Tool is designed to handle only RGB-possible colors, we implemented an *RGB-Clipping* regardless of the selected interpolation color space. Even though, we will show some useful combinations of the different algorithms in Section 6, we want to warn that they partly counteract each other and combinations have not thoroughly been investigated.

### 4.1. Uniformity

A colormap suffices uniformity if the speed of the colormap is constant. With constant speed, the perceived distances between colors are proportional to the distances of the values they represent. Figure 1 shows a simple application of the local and global uniformity optimization.

#### 4.1.1. Local Uniformity

A colormap is locally uniform if its local speed (4) is constant, i.e.,

$$\forall j \in \{1, \dots, n\} : v_j = \frac{d_j}{|t_j - t_{j-1}|} = \bar{v} \quad (7)$$

with the average local speed  $\bar{v} \in \mathbb{R}$  from (5). Local uniformity is one of the few colormap qualities for which algorithms have been suggested in the literature [Lev96, Gre08]. We follow their main idea and will deviate only slightly from their suggested algorithms for simplicity and neither change the colors, nor the number of control points. Local uniformity can be achieved for any colormap through only adapting the positions of the control points to

$$t'_j = \frac{\sum_{i=0}^j d_i}{\sum_{i=0}^n d_i} \quad (8)$$

because then for all  $j$

$$v'_j \stackrel{(4)}{=} \frac{d_j}{|t'_j - t'_{j-1}|} \stackrel{(8)}{=} \frac{d_j}{\left| \frac{\sum_{i=0}^j d_i}{\sum_{i=0}^n d_i} - \frac{\sum_{i=0}^{j-1} d_i}{\sum_{i=0}^n d_i} \right|} = \frac{d_j}{\left| \frac{d_j}{\sum_{i=0}^n d_i} \right|} = \sum_{i=0}^n d_i. \quad (9)$$

This adaption leaves the colors at the control points, the discriminative power, and the order unchanged.

#### 4.1.2. Global Uniformity

A colormap is globally uniform if it has constant global speed (2)

$$\forall i, j \in \{0, \dots, n\} : V_{i,j} = \frac{D_{i,j}}{|t_i - t_j|} = \bar{V} \quad (10)$$

with the average global speed  $\bar{V} \in \mathbb{R}$  from (6). Global uniformity is equivalent to linearity in a Euclidean setting. So if the colors of a given colormap do not happen to lie on a straight line, it cannot be achieved through only adapting the positions of the control points. In a Euclidean setting, the closest globally uniform colormap is the straight line that best approximates the original one, which can be

determined through linear regression. Let  $T \in \mathbb{R}^{m \times 2}$

$$T = \begin{pmatrix} 1 & t_0 \\ \vdots & \vdots \\ 1 & t_m \end{pmatrix} \quad (11)$$

be the matrix containing the control points  $t_0, \dots, t_m$ , and  $X^i = (x_0^i, \dots, x_m^i)^T \in \mathbb{R}^m$  the  $i$ -th coordinate in a Euclidean color space, then we get the parameters  $A^i = (a_0^i, a_1^i)^T$  of the closest line  $l(t) = a_0^i + a_1^i t$  through

$$A^i = (T^T T)^{-1} T^T X^i. \quad (12)$$

It might be useful for an application that the endpoints or one endpoint remain unchanged, for example, if the user changes a subinterval adjacent to a control point whose color will not change. Then, the problem simplifies to only one degree of freedom.

For the customized application using the slider, we interpolate linearly between the control colors  $x_i$  of the original colormap and the corresponding ones on the line  $l(t_i)$ .

### 4.2. Intrinsic Order

Intrinsic order [BTRA18] is satisfied if the perceived distance between two points in a colormap is larger than the distance to any point between them

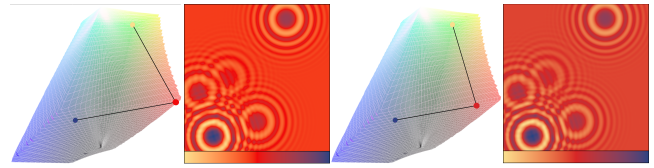
$$\forall i < j < k \in \{1, \dots, n\} : D_{i,j} < D_{i,k} > D_{j,k}. \quad (13)$$

The local version is achieved for  $i = j - 1, k = j + 1$ . It is a special case of the global one and can be treated as such. Figure 2 illustrates that there is an intuitive way to establish intrinsic order, namely by modifying the location of  $c_j$  so that it lies within the intersection area of the two spheres around the colors  $c_i$  and  $c_k$  with the radius  $r = \|c_i - c_k\|$ . The remainder of this section describes how the closest point in that region can be computed geometrically.

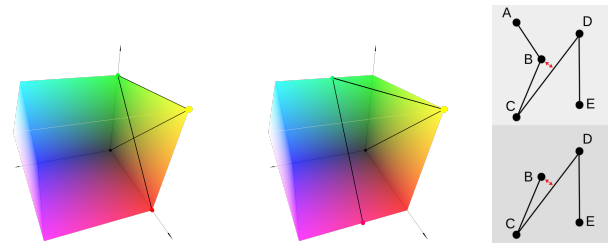
Assume that  $\|c_j - c_k\|$  or  $\|c_j - c_i\|$  is greater than the radius  $r = \|c_i - c_k\|$ . In this case the colors  $c_i, c_j$ , and  $c_k$  violate the intrinsic order and we need to find a new position  $c'_j$  for the color  $c_j$ . The optimal position is the closest point within the intrinsic order area, so that the colors fulfill the order and the change from the color  $c_j$  to  $c'_j$  is as small as possible. For the calculation of this closest position we have to distinguish between the following four cases that are also illustrated in Figure 2. Please note that all algorithms are designed to work in 3D, but since any three points in 3D can be placed in a plane, we can use clearer 2D illustrations. Without loss of generality, let  $\|c_j - c_k\| \geq \|c_j - c_i\|$ .

**Case 1:** If  $\|c_j - c_k\| > r$  and  $\|c_j - c_i\| \leq r$ , the color  $c_j$  is inside the sphere  $B_r(c_i)$ . The line connecting  $c_j$  to the closest point in the intrinsic order region is orthogonal to the surface of the region. Therefore we calculate the intersection points between the line  $\overline{c_j, c_k}$  and the sphere  $B_r(c_k)$ . The closest intersection point to  $c_j$  is located on the surface of the intrinsic order region and is the optimal position  $c'_j$ .

**Case 2:** If  $\|c_j - c_k\| > r$  and  $\|c_j - c_i\| \geq r$ , the color  $c_j$  is located outside of both spheres. We calculate the intersection points between the line  $\overline{c_j, c_i}$  and the sphere  $B_r(c_i)$  and also the intersection points between the line  $\overline{c_j, c_k}$  and the sphere  $B_r(c_k)$ .



**Figure 3:** Local Intrinsic Order Example applied the Langermann test-function [MS05]; the input CMS (left) has a break of the intrinsic order. On the right side is the optimized CMS.



**Figure 4:** Left: Example of a break of the global legend-based order caused by an intersection of the interpolation line (RGB colorspace) and the solution after the application of the global legend-based order optimization. Right: End – Node Repulsive Force Example: In both graphs, the node B is too close to segment  $\overline{CD}$ . If B is not an end node (top), both line-segments  $\overline{AB}$  and  $\overline{CD}$  would be treated and the resulting repulsive force would repel them. For the bottom graph, we need the additional repulsive force between end nodes and the adjoining line segment.

**Case 2a:** If only one of these four calculated intersection points has a smaller distance to  $c_i$  and  $c_k$  than the radius  $r$  and is therefore at the surface of the intrinsic order area, it is the optimum  $c'_j$ .

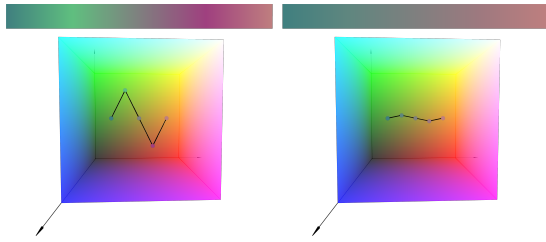
**Case 2b:** If each of the two lines has one intersection point located at the surface of the intrinsic order area, we calculate the distances from  $c_j$  to both intersection points and choose the closest one for the new position  $c'_j$ .

**Case 2c:** If not a single one of the four intersection points lies on the surface of the intrinsic order area, the optimum position is the closest point to  $c_j$  on the intersection circle.

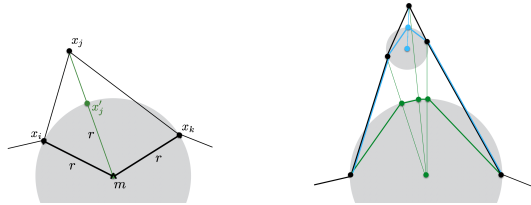
Figure 3 shows an example for local intrinsic order optimization.

### 4.3. Legend-based Order (LBO)

Legend-based order (LBO) [BTS\*18] allows an observer to interpret which colors represent higher values and which colors represent lower values while consulting the color legend. It can be mathematically expressed through invertibility and can be measured through the minimum distance or the minimum speed. For the rest of this section, we describe the algorithms using the speed. In the tool, the users can choose which option they prefer. The distance version pushes points away from each other independent of their respective control points to increase the discriminative power, while the speed-based version will repel points further if they correspond to farther apart values in the data.



**Figure 5:** This figure shows an example for the smoothness optimization by key color adjustment. On the left side a simple example of a zigzag-like colormap with a strong curvature at the peaks. On the right side the solution of our optimization algorithm.



**Figure 6:** Left: construction of the closest point satisfying a given curvature threshold. Right: difference between local (blue) and global (green) intrinsic smoothness.

#### 4.3.1. Local Legend-based Order

Local LBO is violated if consecutive control points  $t_i, \dots, t_j, i \neq j$  have the same color  $x_i = \dots = x_j$ . If an unintentional violation of local LBO occurs, an arbitrarily small offset of any of the involved control colors can fix this invertibility issue. However, this will not necessarily improve the overall perceptual quality of the colormap if the offset is too small to be noticeable by the human eye.

The LBO can be measured through the minimum local speed. If two neighboring points are too close together, we push them apart using a force-based algorithm. For the implementation, we were inspired by the force-directed graph concept by Fruchterman and Reingold [FR91]. We use the key colors as nodes to create a graph. For all continuous sections of the CMS, we connect the nodes accordingly with an edge. We use two forces. The first one minimizes the distortion of the colormap by attracting each node to its original position. The second one increases the minimal speed by repelling points that are too close to each other.

The ratio of the powers of both forces is steered by the user through the slider thereby setting the degree of the LBO optimization. The slider position is proportional to the minimal speed. The left boundary corresponds to a speed of zero for reference. The right boundary is cut off at the maximal speed the longest straight line can have in the selected color space to prevent the user from exceeding the region where minimal speed really corresponds to the concept of LBO in a meaningful way.

#### 4.3.2. Global Legend-based Order

A break in global LBO is produced by a self-intersecting path through color space. It can either come from non-adjacent control

points having the same color or the interpolation resulting in self-intersection between control points.

To optimize the global LBO, we also use a modified forced-directed graph algorithm. Similar to the local LBO algorithm, we create a node for all key colors and connect successive key colors with edges.

The idea of an origin force that draws keys to their original locations is the same as the local version, but the repulsive force is different with respect to the emergence of the power. For the global LBO, we have to check the colormap for intersections or rather for close progressions of the interpolation path. Therefore, the repulsive force should not only consider the color (node) positions as at the local version, but also the edges of the graph. If the algorithm detects such a break of the order, the involved path segments need to be repelled in the opposite direction (see Figure 4).

We consider only non-adjacent edges and use the speed between the closest points of both segments, which is determined using basic geometry [KHK89], for the determination of the repulsive force.

#### 4.4. Discriminative Power

Discriminative power (DP) is described as the number of just noticeable differences (JNDs) that are traversed by the colormap. We distinguish between a local and a global form of DP. The local DP can be achieved through neighboring points repelling each other and the global DP through all points repelling each other. With our forced-directed graph variations of the legend-based order optimization using speed or distance, we have already created a particular algorithm to improve local and global DP. The difference is that we do not limit the slider to small speeds or distance.

#### 4.5. Intrinsic Smoothness

So far, we improved the measures suggested by Bujack et al. [BTS\*18] in our algorithms, but we have not dealt with smoothness up to now.

Smoothness is especially challenging, because there are many criteria that can influence how smooth a colormap is perceived. Areas of a colormap can produce the impression of sharp bends due to varying speed [PZJ82b, PZJ82a, LH92, Lev96, RT98, Rhe00, ZM06, BTI07], actual bends in the path of the colormap [RO86, Pha90, Mor, ZH16, WVVWVDL08], changes in visual importance [BSM\*15, LB04, MJSK15b, Rhe00, CYG02, HB13], or borders between areas of colors belonging to a common name group [SP11, LFK\*13, HS12, Sch15]. We exclude the differences in speed from our analysis here, because we have already treated this factor under uniformity in Section 4.1. We further assume that the influence of the visual importance and the color names on the perceived distances is encoded in the metric of the underlying color space. Even if that is not achieved perfectly in an easily accessible way, there is promising work in that direction [GM19] so that we can expect the quality of the results to increase automatically with the development of better colorspace.

What is left, we call intrinsic smoothness, by which we mean the actual bends in the path of the colormap. We can measure intrinsic

smoothness through the curvature of the curve of a colormap using differential geometry [TR06, Küh15], as shown in Figure 5.

#### 4.5.1. Key Color Smoothness

There are various approaches to extend differential geometry of curves to the discrete setting [Cha00, GDP\*06, BSSZ08]. We will use a notation that is related to the one by Sauer [Sau70].

The tangent vector is given for  $i = 1, \dots, n$  by the backward finite differences

$$\bar{T}_i = \frac{x_i - x_{i-1}}{t_i - t_{i-1}}. \quad (14)$$

We define the curvature angle for  $i = 1, \dots, n - 1$  using the displacement between two consecutive tangent vectors by

$$\tilde{\kappa}_i = \angle(T_i, T_{i+1}) = \arccos(\langle T_i, T_{i+1} \rangle) \quad (15)$$

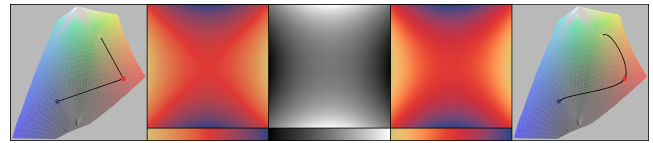
and the discrete curvature as the sine of the curvature angle scaled by the average length of the two adjacent segments

$$\kappa_i = \frac{\sin(\tilde{\kappa}_i)}{\frac{1}{2}(t_{i+1} - t_{i-1})} = \frac{\sqrt{1 - \langle T_i, T_{i+1} \rangle^2}}{\frac{1}{2}(t_{i+1} - t_{i-1})}. \quad (16)$$

We choose to scale with the average of the two adjacent tangent lengths, because the curvature is a measure of how much two adjacent tangents change, which is associated with a vertex and not a single edge. The core idea is to remove bends in the path by reducing the curvature at any point where a threshold set through a slider is exceeded and to increase it where a second threshold is succeeded.

If the curvature  $\kappa_j$  between three points  $x_i, x_j, x_k, i < j < k$  exceeds the threshold, we relocate  $x_j$  to the closest color satisfying the provided curvature. The curvature of a differentiable curve is the inverse of the radius of the osculating circle. Therefore, we know from the inscribed angle theorem that the closest point to  $x_j$  that satisfies the prescribed curvature lies on the circle through  $x_i$  and  $x_k$  with radius  $r_j = 1/\kappa_j$  in the plane  $p_1$  through  $x_i, x_j, x_k$ . It can be calculated as the intersection between the circle and the line connecting  $x_j$  to the center of the circle, as in Figure 6. To get the circle center  $m$  in the plane through  $x_i, x_j, x_k$ , we calculate the middle point  $x_s$  between the points  $x_i$  and  $x_k$ . With  $x_s$  and the direction vector  $\vec{c}_i, \vec{c}_k$ , we construct an orthogonal plane  $p_2$ . Imagine that we have two spheres with the centers  $x_i$  and  $x_k$  with the radius  $r_j$ . Then, the intersection circle of these two spheres would lie in the plane  $p_2$ . As the point  $m$  has to be on the intersection circle of both spheres, we know that  $m$  lies in our plane  $p_2$  as well as on the plane  $p_1$ . Next, we calculate the intersection line  $g$  between  $p_1$  and  $p_2$ . With  $g$ , the distance  $r_j$ , one of the points  $x_i$  or  $x_k$ , and the Pythagorean theorem, we can determine a quadratic equation. With the solution of this equation, we get none, one, or two possible solutions for our searched point  $m$ . In the case of two possible solutions, we choose the point that is further away from  $x_j$ . If there is no solution for  $m$ , we break the optimization for  $x_i, x_j$ , and  $x_k$ . Once a point has been moved, its neighbors need to be checked again.

In exactly the same way, a point is moved away onto the closest location that satisfies the threshold chosen by the user to limit the curvature angle from below if it violates that threshold. For local



**Figure 7:** *Intrinsic Smoothness Example:* The figure shows the interpolation types (left: linear, right: spline). The saddle-point visualizations demonstrate the differences (grey-scaled for comparison); The linear interpolation causes a more visible red "X"-shape. The spline interpolation result in a smoother colour shift.

intrinsic smoothness, we only consider consecutive points  $i + 1 = j = k - 1$ . The global version considers all variations of  $i < j < k$ .

#### 4.5.2. Smooth Interpolation (Splines)

For an alternative implementation of local intrinsic smoothness considering the interpolation, we decided to follow an alternative way to reduce the curvature that maintains the existing control points and colors by only changing the path between the key points, i.e., leaving control points  $t_i$  and colors  $c_i$  unchanged. This method cannot achieve zero curvature, but may be preferred because of its low distortion of the original colormap. In our implementation, we follow the suggestion of Pham [Pha90] and allow the user to choose a spline [CR74] instead of a piece-wise linear polygon chain as their means of interpolation between the given key points. That way, all original key points are maintained, and no additional key points need to be stored. Furthermore, the bending energy, more precisely its approximation by the integral over the second derivative, is minimized [FF02, pp.167–169].

For the realization of the spline interpolation in the CCC-Tool, we implemented the Catmull Rom Splines [CR74] method, which is part of the family of cubic Hermite splines and has  $C^1$  continuity. The ability of this method for local control and interpolation is ideal for editing CMS in the tool. Figure 7 shows an example for spline interpolation compared to linear interpolation.

### 5. Evaluation

It is not easy to determine in a formal evaluation how much better a colormap is after optimization, because our algorithms improve a colormap only w.r.t. a specific mathematical design rule each, while the overall quality of a colormap depends on a myriad of parameters, like the visualization task, data, intended audience, display, or visualization goal [BRT95, RTB96, Rhe00, TFS08]. Just because a colormap is made uniform does not mean that it performs better in every way [War88]. A user can even use the algorithms to make a good colormap worse.

Fig. 1 gives an example of a mathematical evaluation of the algorithm generating uniformity. The **C-Row** shows images of the input colormap and its improved versions applied to the Langermann test-function [MS05]. The **R-Row** shows the function report of the CCC-Tool, which creates a subtraction field from a normalized gradient field of the test-function values and a normalized color difference field to show uniform, under- or overestimated

areas [NCB\*20]. Panel **C1** shows the visualizations for a non-uniform colormap with a yellow-red gradient. The corresponding report image (**R1**) indicates that the colormap only inaccurately reflects the values of the test-function. Panel **C2** shows the colormapping after a local uniform optimization, yielding an improved but not perfect representation of the test-function shapes (**R2**). Compared with **C1**, we can see that the original colormap gives a wrong impression of test-function shapes. As well **R2** shows a reduction of areas with erroneous perception of the test values. Depending on the actual function and the resolution used, large color differences can occur between two adjacent pixels. Local uniformity is struggling to deliver a uniform transition between such pixels. The globally uniform algorithm is able to perfectly represent the shape of the underlying function **C3** through its color difference being proportional to value difference **R3**.

This report shows that the colormap is indeed more uniform after the application of the algorithm. Similar tests can be generated for all algorithms. If used right, they all improve the mathematical property they were designed to improve. However, this more technical type of evaluation would not produce significant new insight and exceed the page limit.

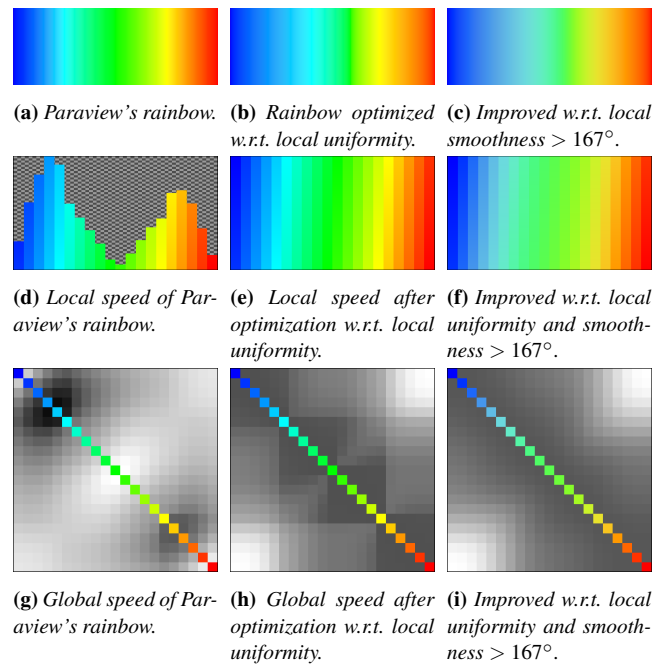
Instead, we provide feedback from two users familiar with the tool, and in the following section we present two application scenarios. To get the feedback, we asked Francesca Samsel and Michael Böttinger to try our optimization algorithm and write their responses in emails.

Francesca Samsel is an artist specialized in working with application scientists to create custom colormaps that are tuned to specific data and tasks needed to address their specific science questions. Please refer to her team's work and colormaps that were designed in the CCC-Tool for a large amount of example applications, like topographical data [ZSN20], ocean currents [WSR\*20], or the chemical properties of water masses under Antarctica's ice sheets [HSB\*20]. About her experience with the tool, she writes in an informal interview: "We use CCC-Tool exclusively as it is the only colormap creation tool with the fidelity of control required. The Optimization Function saves hours of time producing smoother maps with equalized contrast distribution. Previously the first version of a custom colormap required four to five hours to construct. Given that we bring the scientists several choices, it is a lengthy process. Building maps in the CCC-Tool requires 20 to 30 minutes."

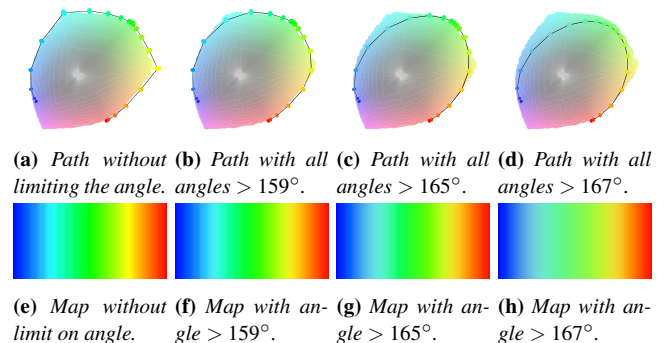
Following a direct quote from Michael Böttinger, a scientist at the German Climate Computing Center (DKRZ): "The quality of a colormap depends on a multitude of parameters such as specific tasks, communication goals, audience, etc. In practice we often have to find a compromise between all these criteria and hence the different optimization options in order to create meaningful visualizations. The CCC-Tool helped us to better understand and practically apply the different and partly contradicting optimization strategies."

## 6. Application

In Figures 1, 3, 4, 6, and 7, we already have given little synthetic examples for each algorithm individually. In this section, we will



**Figure 8:** Improving Paraview's rainbow w.r.t. local uniformity and smoothness. Top: the colormap. Middle: local speed in DIN99. Bottom: global speed in DIN99 (black high, white low).



**Figure 9:** Different stages while improving Paraview's rainbow w.r.t. smoothness in DIN99.

show how our algorithms can help to improve colormaps in practical applications.

### 6.1. Improvement of the Rainbow Colormap

The rainbow is one of the most popular colormaps even though it has received broad criticism. It has regions with very low discriminative power that can hide features around green, and mach bands around yellow and cyan that can introduce artifacts into a visualization [BT107, Mor09, RO86, RT98, WF80].

The analysis part of the CCC-Tool shows that the rainbow already suffices legend-based order and local intrinsic order and has a pretty good discriminative power, but as pointed out in previous



work has deficiencies in uniformity and smoothness, which we will improve; see Figure 8.

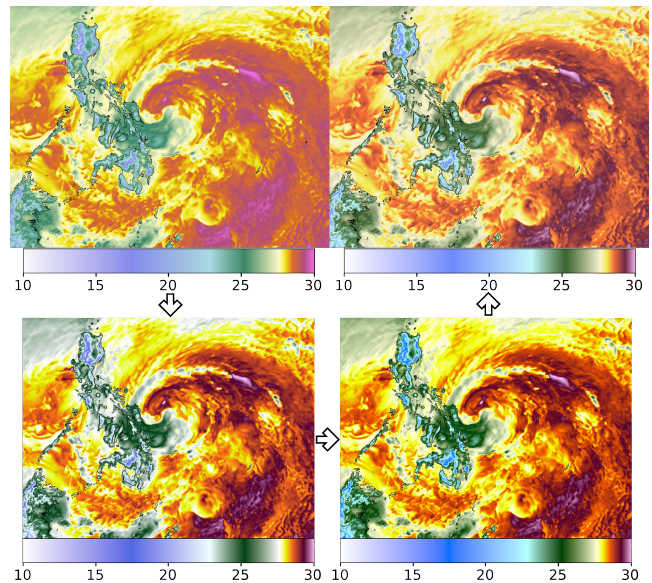
Since improving the colormap w.r.t. local uniformity does not influence the colors, we can push the slider all the way to the right to assume the optimum without significantly changing the colormap. Figure 8 middle shows that the local speed is constant now across the colormap. This contracts the big green region in the middle that had very low discriminative power. Still, the mach bands around yellow and cyan remain. A closer look into the path plot reveals that these are not caused by a lack of uniformity but by a local non-smooth bend in the curve through colorspace. We improve the colormap w.r.t. local smoothness. Figure 9 top shows how the path of the colormap becomes more smooth as we increase the minimum angle. Putting the slider all the way to the right would produce a straight line, which is not the desired outcome. We find a minimum angle of  $167^\circ$  a satisfactory tradeoff to smooth out the mach bands without distorting the colormap too much into desaturation. Figure 8 right shows that the global smoothness becomes more uniform as the path becomes closer to a circle. We used linear interpolation in DIN99 and ParaView's rainbow colormap sampled at 20 points.

## 6.2. Improvement of a Custom Colormap Developed for a Tropical Storm

In practical applications, colormaps are occasionally more complex than simple gradients from one color to another or typical divergent colormaps. Often they are customized and tweaked to put focus on a specific part of the data range. However, handcrafting colormaps within the process of interactive visualization is a subjective process that may result in colormaps that are not optimal with respect to the design criteria discussed above. We picked an example where nonlinear mapping is used to visually enhance a small data range by local high speed. This colormap contradicts some of the design criteria and might not be optimal with regard to others. For example, analysis of the colormap shows that its discriminative power (DP) is not yet optimized. Here we demonstrate with a real world use case how automatic optimization impacts a complex colormap.

### 6.2.1. Simulation Data

For this work we used a retrospective simulation of the Typhoon Haiyan that hit the Philippines in November 2013, one of the strongest typhoons ever observed, carried out by the Helmholtz-Center Geesthacht (HZG). The model used was a regional high-resolution version of the atmosphere model COSMO-CLM [RWH08, vSFG\*17]. For our tests, we used the air temperature at 2-m height around the time when the storm made landfall at the coast of the Philippines. Figure 10 A shows a visualization of the temperature produced with ParaView [AGL05] with a colormap that covers the full data range and, at the same time, resolves the most important features found in the data. In the region of interest and for the time step chosen, the 2-m height temperature ranges from  $10^\circ\text{C}$  to about  $30^\circ\text{C}$ . As a consequence of the land orography, low temperatures are found on land at high altitudes, while particularly high temperatures of up to  $30^\circ\text{C}$ , related to the warm moist air of the storm system, are mostly found over the ocean. Due to cooling by strong rainfall, areas with cooler temperatures of about



**Figure 10:** This figure illustrates different colormap optimization methods consecutively applied to a visualization of Typhoon Haiyan. This application case is discussed in Section 6.2. The visualizations show the temperature at 2-m height for November 7, 2013, at 19:18 as simulated with the regional model COSMO-CLM. **A:** The original colormap was designed during interactive visualization of the data in ParaView in order to emphasize the important features of the storm. **B:** Application of method to increase the local discriminative power. **C:** One negative effect of the optimization shown in B is the ambiguity caused by several occurrences of key colors that are close to white. Therefore, we optimized this colormap with respect to its global legend-based order. **D:** The colormap of C is quite “loud”, i.e., very strongly saturated. In order to derive a more harmonic colormap, we applied the method to optimize the intrinsic order.

$25^\circ\text{C}$  develop in a semi-circular front near the center of the storm, here visible as green tones at the eastern shore of the Philippines.

### 6.2.2. Colormap Analysis

We started with a colormap interactively developed by domain scientists with the colormap editor of ParaView. This colormap was exported in json-format and then imported to CCC-Tool. Once a CMS is imported and opened, its characteristics are visualized in three different linked views: a 3D path plot shows the keys and the connecting path elements in the chosen color space, a 2D projection of the keys and the path.

The initial CMS already shows noticeable color and lightness variations for the data range above  $27.5^\circ\text{C}$  in order to better resolve the data range that is related to the typhoon for this meteorological and geographical setting. This is reflected in the short distances between the keys for this data range.

The jagged uppermost curve (the lightness curve) illustrates the concept of lightness variations to increase the contrast between keys and hence enhance the discriminative power within the CMS.

The darkest keys of the initial CMS only use a lightness of only about 45 (Lab), which leaves room for further optimizations.

### 6.2.3. Effects of Optimization

To illustrate the effects of different colormap optimization algorithms available in CCC-Tool on the imported CMS, we consecutively applied some of them, exported the modified CMSs using the json format and re-imported them in ParaView. The optimization workflow we applied is not necessarily the only or the simplest one that would achieve the same or at least a similar result; here we intend to illustrate the choice of different optimization methods and their effects to achieve specific goals.

First, our goal is to increase the DP of the CMS in order to resolve more detail in the temperature field. For the DP optimization we chose the “distance” method and relatively strong settings of 100 and 0.7 for its two parameters. Particularly in the lower part of the CMS, which is characterized by larger distances between the keys, the variations of the lightness curve are much more pronounced in comparison with that of the original CMS.

Accordingly, the visualization based on the DP-optimized CMS (Figure 10 B) shows a strongly enhanced brightness contrast compared with the visualization using the original CMS (Figure 10 A) and thus more discriminative power especially for the data range between 10 and 23°C. The mapping between the data values and the position of the keys and hence the intended emphasis on the data range between 27.5 and 29.5°C is maintained, while more discriminable colors are available and hence many more small-scale patterns corresponding to slight variations in the temperature range of interest can now be resolved. However, at the same time the DP optimization introduced some ambiguity as the resulting CMS includes now several quite similar keys that are nearly white.

To overcome this potential problem, we decided to further apply a global legend-based order optimization. This method is useful to prevent close colors within the CMS. Again, we used the “distance” setting, and we used parameter values of 100 and 0.2. The visualization in Figure 10 C specifically shows the desired changes in the bright parts of the colormap while the discriminative power especially in the data range of interest is maintained. However, at the same time, the color tones in the bluish part of the CMS are modified and exhibit higher saturation levels. Generally, the resulting visualization is quite colorful and contrast rich; but, on the other hand, it is not harmonious and is visually loud. Some of the outstanding key colors could be interpreted as regions of danger; areas simply show regions with warm and humid air.

Aiming at a subtler and more harmonious colormap design, for example, with respect to its use for communication to the public, we finally optimized the CMS of Figure 10 C with respect to its global intrinsic order. This optimization method strives to move the color of a key located between other keys towards the color between those keys. Here we used a parameter setting of 0.3. The final result is shown in Figure 10 D. The small-scale details in the temperature field indicating the structure of the typhoon are still quite visible, although the colorscale and contrast are much smoother compared to Figure 10 B and C.

## 7. Conclusion

With this paper, we present algorithms to address an automated optimization of colormaps based on different evaluation criteria. A summary can be found in Table 1. Colormapping is one of the most widely used visualization methods. Many of the colormaps used in practice, however, are not optimal with respect to the intended visualization goals. With this work on automatic colormap optimization, we aim at supporting the user in gaining a better understanding of the underlying theoretical aspects of colormap design and the evaluation criteria applied within the community. By making automatic colormap optimization accessible through a web-based tool, we hope to advance this research area and, at the same time, support visualization designers of various application areas in their work.

For the implementation, we decided to use the web-based CCC-Tool (Version 0.9) to offer a readily accessible application for our automatic optimization. In Section 4, we describe the design rules we used for uniformity, intrinsic order, legend-based order, discriminative power, and smoothness with specific consideration of this specific criterion.

Although the tool technically allows users to apply multiple optimization methods, we here focused on the impact of only one method at a time. Any optimization with respect to one criterion obviously changes the initial condition for a second optimization according to another criterion. Since the criteria partly contradict each other, the result of multi-criteria optimizations are hard to predict; the resulting colormap could potentially be worse relative to the single criterion than the initial colormap. For this reason, we recommend care in using multiple methods.

The examples in Section 4.1-4.5 and the application examples in section 6 demonstrate that either single-criterion optimizations as well as combinations of them can be useful and improve colormaps with respect to different criteria.

A limitation in the current implementation is that the optimization algorithms do not consider color semantics, brightness, or saturation, which are also important for evaluation of colormaps. Furthermore, this work is limited to the Euclidean space, so the optimization cannot use metric information from DE94 or CIEDE2000 for calculations in the Lab color space.

In future work, we envision the modification of our single-criteria methods concerning optimization in non-Euclidean color spaces for the integration of color metrics (e.g., CIEDE2000) in our methods. Furthermore, we need to better understand interactions between different evaluation criteria in order to develop meaningful multi-criteria optimizations.

## Acknowledgements

Research presented in this article was partly supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20200512ECR. We thank F. Feser of the Helmholtz-Center Geesthacht for the permission to use the COSMO-CLM data of Typhoon Haiyan. Open access funding enabled and organized by Projekt DEAL. [Correction added on 29 June 2022, after first online publication: Projekt Deal funding statement has been added.]

## References

- [AGL05] AHRENS J., GEVECI B., LAW C.: 36 - paraview: An end-user tool for large-data visualization. In *Visualization Handbook*, Hansen C. D., Johnson C. R., (Eds.). Butterworth-Heinemann, Burlington, 2005, pp. 717 – 731. URL: <https://www.sciencedirect.com/science/article/pii/B9780123875822500381>, doi:<https://doi.org/10.1016/B978-012387582-2/50038-1>. 2, 9
- [BRT95] BERGMAN L. D., ROGOWITZ B. E., TREINISH L. A.: A rule-based tool for assisting colormap selection. In *Proceedings of the 6th conference on Visualization'95* (1995), IEEE Computer Society, p. 118. 2, 7
- [BSM\*15] BERNARD J., STEIGER M., MITTELSTÄDT S., THUM S., KEIM D., KOHLHAMMER J.: A survey and task-based quality assessment of static 2D colormaps. In *SPIE/IS&T Electronic Imaging* (2015), International Society for Optics and Photonics, pp. 93970M–93970M. 6
- [BSSZ08] BOBENKO A. I., SULLIVAN J. M., SCHRÖDER P., ZIEGLER G.: *Discrete differential geometry*. Springer, 2008. 7
- [BTI07] BORLAND D., TAYLOR II R. M.: Rainbow color map (still) considered harmful. *IEEE computer graphics and applications* 27, 2 (2007), 14–17. 6, 8
- [BTRA18] BUJACK R., TURTON T. L., ROGERS D., AHRENS J.: Ordering Perceptions about Perceptual Order. In *IEEE Scientific Visualization Conference (SciVis) Short Papers* (2018), IEEE. 5
- [BTS\*18] BUJACK R., TURTON T. L., SAMSEL F., WARE C., ROGERS D. H., AHRENS J.: 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 (Jan 2018), 923–933. doi:[10.1109/TVCG.2017.2743978](https://doi.org/10.1109/TVCG.2017.2743978). 1, 2, 3, 5, 6
- [CBW\*12] CHILDS H., BRUGGER E., WHITLOCK B., MEREDITH J., AHERN S., PUGMIRE D., BIAGAS K., MILLER M., HARRISON C., WEBER G., KRISHNAN H., FOGAL T., SANDERSON A., GARTH C., BETHEL E. W., CAMP D., RUBEL O., DURANT M., FAVRE J., NAVRATIL P.: Visit: An end-user tool for visualizing and analyzing very large data. *High Performance Visualization-Enabling Extreme-Scale Scientific Insight* (10 2012), 357–372. 2
- [Cha00] CHAN R.: *Discrete Curves and Surfaces*. PhD Dissertation, Technical University Berlin, Germany, 2000. 7
- [CR74] CATMULL E., ROM R.: A class of local interpolating splines. In *Computer Aided Geometric Design*, BARNHILL R. E., RIESENFELD R. F., (Eds.). Academic Press, 1974, pp. 317 – 326. 7
- [CYG02] CAMGÖZ N., YENER C., GÜVENÇ D.: Effects of hue, saturation, and brightness on preference. *Color Research & Application* 27, 3 (2002), 199–207. 6
- [EAM11] EISEMANN M., ALBUQUERQUE G., MAGNOR M.: Data driven color mapping. In *Proceedings of EuroVA: International Workshop on Visual Analytics, Bergen, Norway* (2011), Citeseer. 3
- [FF02] FARIN G. E., FARIN G.: *Curves and surfaces for CAGD: a practical guide*, 5 ed. Morgan Kaufmann, 2002. 7
- [FR91] FRUCHTERMAN T. M. J., REINGOLD E. M.: Graph drawing by force-directed placement. *Software: Practice and Experience* 21, 11 (1991), 1129–1164. 6
- [FWD\*17] FANG H., WALTON S., DELAHAYE E., HARRIS J., STORCHAK D. A., CHEN M.: Categorical colormap optimization with visualization case studies. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 871–880. doi:[10.1109/TVCG.2016.2599214](https://doi.org/10.1109/TVCG.2016.2599214). 2
- [GDP\*06] GRINSPUN E., DESBRUN M., POLTHIER K., SCHRÖDER P., STERN A.: Discrete differential geometry: an applied introduction. *ACM SIGGRAPH Course* 7 (2006). 7
- [GM19] GRIFFIN L. D., MYLONAS D.: Categorical colour geometry. *PLoS one* 14, 5 (2019), e0216296. 6
- [Gre08] GRESH D. L.: *Self-Corrected Perceptual Colormaps*. Tech. rep., IBM, 2008. RC24542 (W0804-104). 1, 2, 4
- [HB03] HARROWER M., BREWER C.: Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal* 40, 1 (2003), 27–37. 2
- [HB11] HARROWER M., BREWER C. A.: *ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps*. Wiley-Blackwell, 2011, ch. 3.8, pp. 261–268. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470979587.ch34>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470979587.ch34>, doi:[10.1002/9780470979587.ch34](https://doi.org/10.1002/9780470979587.ch34). 2
- [HB13] HARROWER M., BREWER C. A.: Colorbrewer.org: an online tool for selecting colour schemes for maps. *The Cartographic Journal* (2013). 6
- [HMO06] HUERTAS R., MELGOSA M., OLEARI C.: Performance of a color-difference formula based on OSA-UCS space using small-medium color differences. *JOSA A* 23, 9 (2006), 2077–2084. 2
- [HS12] HEER J., STONE M.: Color naming models for color selection, image editing and palette design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 1007–1016. 6
- [HSB\*20] HERMAN B., SAMSEL F., BARES A., JOHNSON S., ABRAM G., KEEFE D. F.: Printmaking, puzzles, and studio closets: Using artistic metaphors to reimagine the user interface for designing immersive visualizations, 2020. arXiv:[2010.08859](https://arxiv.org/abs/2010.08859). 8
- [Hun07] HUNTER J. D.: Matplotlib: A 2d graphics environment. *Computing In Science & Engineering* 9, 3 (2007), 90–95. doi:[10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55). 2
- [Jac13] JACOMY M.: iwanthue, 2013. URL: <https://medialab.github.io/iwanthue/>. 2
- [KHK89] KRÄMER H., HÖWELMANN R., KLEMISCH I.: *Analytische Geometrie und lineare Algebra*. Diesterweg, 1989. 6
- [Kov15] KOVESI P.: Good colour maps: How to design them. *CoRR* abs/1509.03700 (2015). URL: <http://arxiv.org/abs/1509.03700>, arXiv:[1509.03700](https://arxiv.org/abs/1509.03700). 2
- [KRPC00] KALVIN A. D., ROGOWITZ B. E., PELAH A., COHEN A.: Building perceptual color maps for visualizing interval data. In *Human Vision and Electronic Imaging V* (2000), vol. 3959, International Society for Optics and Photonics, pp. 323–336. 2
- [Küh15] KÜHNEL W.: *Differential Geometry: Curves - Surfaces - Manifolds*. Student Mathematical Library. American Mathematical Society, 2015. URL: <https://books.google.com/books?id=qNBYCwAAQBAJ>. 7
- [LB04] LIGHT A., BARTLEIN P. J.: The end of the rainbow? Color schemes for improved data graphics. *Eos* 85, 40 (2004), 385–391. 6
- [LCL06] LUO M. R., CUI G., LI C.: Uniform colour spaces based on ciecam02 colour appearance model. *Color Research & Application* 31, 4 (2006), 320–330. 2
- [Lev96] LEVKOWITZ H.: Perceptual steps along color scales. *International Journal of Imaging Systems and Technology* 7, 2 (1996), 97–101. 4, 6
- [LFK\*13] LIN S., FORTUNA J., KULKARNI C., STONE M., HEER J.: Selecting semantically-resonant colors for data visualization. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 401–410. 6
- [LH92] LEVKOWITZ H., HERMAN G. T.: The design and evaluation of color scales for image data. *IEEE Computer Graphics and Applications* 12, 1 (1992), 72–80. 2, 6
- [ME094] MAHY M., EYCKEN L., OOSTERLINCK A.: Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV. *Color Research & Application* 19, 2 (1994), 105–121. 2

- [MJSK15a] MITTELSTÄDT S., JÄCKLE D., STOFFEL F., KEIM D. A.: Colorcat: Guided design of colormaps for combined analysis tasks. In *Proc. of the Eurographics Conference on Visualization (EuroVis 2015: Short Papers)* (2015), vol. 2. 2
- [MJSK15b] MITTELSTÄDT S., JÄCKLE D., STOFFEL F., KEIM D. A.: ColorCAT: Guided Design of Colormaps for Combined Analysis Tasks. In *Eurographics Conference on Visualization (EuroVis) - Short Papers* (2015), Bertini E., Kennedy J., Puppo E., (Eds.), The Eurographics Association, pp. 115–119. 6
- [Mor] MORELAND K.: <http://www.kennethmoreland.com/color-maps/>, accessed November 2016. 6
- [Mor09] MORELAND K.: Diverging color maps for scientific visualization. In *International Symposium on Visual Computing* (2009), Springer, pp. 92–103. 2, 8
- [MS05] MOLGA M., SMUTNICKI C.: Test functions for optimization needs. 3, 5, 7
- [MSK14] MITTELSTÄDT S., STOFFEL A., KEIM D. A.: Methods for compensating contrast effects in information visualization. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 231–240. 3
- [NCB\*20] NARDINI P., CHEN M., BUJACK R., BÖTTINGER M., SCHEUERMANN G.: A testing environment for continuous colormaps. *IEEE Transactions on Visualization and Computer Graphics* (2020), 1–1. doi:10.1109/TVCG.2020.3028955. 8
- [NCS\*19] NARDINI P., CHEN M., SAMSEL F., BUJACK R., BÖTTINGER M., SCHEUERMANN G.: The making of continuous colormaps. *IEEE Transactions on Visualization and Computer Graphics* (2019), doi:10.1109/TVCG.2019.2961674. 2, 4
- [oI04] ON ILLUMINATION I. C.: *Colorimetry*. CIE technical report. Commission internationale de l’Eclairage, CIE Central Bureau, 2004. URL: <https://books.google.com/books?id=P1NkAAAACAAJ.2>
- [Pha90] PHAM B.: Spline-based color sequences for univariate, bivariate and trivariate mapping. In *Proceedings of the 1st conference on Visualization '90* (1990), IEEE Computer Society Press, pp. 202–208. 1, 2, 6, 7
- [Piz81] PIZER S. M.: Intensity mappings to linearize display devices. *Computer Graphics and Image Processing* 17, 3 (1981), 262–268. 1
- [PZJ82a] PIZER S. H., ZIMMERMAN J. B., JOHNSTON R. E.: Concepts of the display of medical images. *IEEE Transactions on Nuclear Science* 29, 4 (1982), 1322–1330. 6
- [PZJ82b] PIZER S. M., ZIMMERMAN J. B., JOHNSTON R. E.: Contrast transmission in medical image display. In *1st International Symposium on Medical Imaging and Image Interpretation* (1982), International Society for Optics and Photonics, pp. 2–9. 6
- [RFB16] REINECKE K., FLATLA D. R., BROOKS C.: Enabling designers to foresee which colors users cannot see. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 2693–2704. 3
- [Rhe00] RHEINGANS P. L.: Task-based color scale design. In *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making* (2000), International Society for Optics and Photonics, pp. 35–43. 2, 6, 7
- [RKPC99] ROGOWITZ B. E., KALVIN A. D., PELAH A., COHEN A.: Which trajectories through which perceptually uniform color spaces produce appropriate colors scales for interval data? In *7th Color and Imaging Conference* (1999), Society for Imaging Science and Technology, pp. 321–326. 2
- [RNAK18] REDA K., NALAWADE P., ANSAH-KOI K.: Graphical perception of continuous quantitative maps: The effects of spatial frequency and colormap design. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI '18, Association for Computing Machinery. URL: <https://doi.org/10.1145/3173574.3173846>, doi:10.1145/3173574.3173846. 3
- [RO86] ROBERTSON P. K., O’CALLAGHAN J. F.: The generation of color sequences for univariate and bivariate mapping. *IEEE Computer Graphics and Applications* 6, 2 (1986), 24–32. 1, 2, 6, 8
- [RT98] ROGOWITZ B. E., TREINISH L. A.: Data visualization: the end of the rainbow. *IEEE spectrum* 35, 12 (1998), 52–59. 2, 6, 8
- [RTB96] ROGOWITZ B. E., TREINISH L. A., BRYSON S.: How not to lie with visualization. *Computers in Physics* 10, 3 (1996), 268–273. 2, 7
- [RWH08] ROCKEL B., WILL A., HENSE A.: The regional climate model cosmo-clm (cclm). *Meteorologische Zeitschrift* 17, 4 (08 2008), 347–348. URL: <http://dx.doi.org/10.1127/0941-2948/2008/0309>, doi:10.1127/0941-2948/2008/0309. 9
- [Sau70] SAUER R.: *Differenzgeometrie*. Springer, 1970. 7
- [Sch15] SCHLOSS K. B.: Color preferences differ with variations in color perception. *Trends in cognitive sciences* 19, 10 (2015), 554–555. 6
- [SKR18] SAMSEL F., KLAASSEN S., ROGERS D. H.: Colormoves: Real-time interactive colormap construction for scientific visualization. *IEEE Computer Graphics and Applications* 38, 1 (Jan 2018), 20–29. doi:10.1109/MCG.2018.011461525. 2
- [SMS07] SILVA S., MADEIRA J., SANTOS B. S.: 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* (2007), IEEE, pp. 943–950. 2
- [SP11] SCHLOSS K. B., PALMER S. E.: Aesthetic response to color combinations: preference, harmony, and similarity. *Attention, Perception, & Psychophysics* 73, 2 (2011), 551–571. 6
- [SSM11] SILVA S., SANTOS B. S., MADEIRA J.: Using color in visualization: A survey. *Computers & Graphics* 35, 2 (2011), 320–333. 2
- [SSS14] STONE M., SZAFIR D. A., SETLUR V.: An engineering model for color difference as a function of size. In *Color and Imaging Conference* (2014), vol. 2014, Society for Imaging Science and Technology, pp. 253–258. 3
- [SWS19] SMART S., WU K., SZAFIR D. A.: Color crafting: Automating the construction of designer quality color ramps. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 1215–1225. 2
- [SwTS05] SCHULZE-WOLLGAST P., TOMINSKI C., SCHUMANN H.: Enhancing visual exploration by appropriate color coding. In *Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)* (2005), pp. 203–210. 2
- [Sza18] SZAFIR D. A.: Modeling color difference for visualization design. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan 2018), 392–401. doi:10.1109/TVCG.2017.2744359. 2
- [Taj83] TAJIMA J.: Uniform color scale applications to computer graphics. *Computer Vision, Graphics, and Image Processing* 21, 3 (1983), 305–325. 2
- [TFS08] TOMINSKI C., FUCHS G., SCHUMANN H.: Task-driven color coding. In *2008 12th International Conference Information Visualisation* (2008), IEEE, pp. 373–380. 2, 3, 7
- [TR06] TOPONOGOV V., ROVENSKI V.: *Differential Geometry of Curves and Surfaces: A Concise Guide*. Birkhäuser Boston, 2006. URL: <https://books.google.com/books?id=uPoC8gpoPiEC.7>
- [vSFG\*17] VON STORCH H., FESER F., GEYER B., KLEHMET K., LI D., ROCKEL B., SCHUBERT-FRISIUS M., TIM N., ZORITA E.: Regional reanalysis without local data: Exploiting the downscaling paradigm. *Journal of Geophysical Research: Atmospheres* 122, 16 (2017), 8631–8649. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016JD026332>, arXiv: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2016JD026332>, doi:10.1002/2016JD026332. 9
- [War88] WARE C.: Color sequences for univariate maps: Theory, experiments and principles. *IEEE Computer Graphics and Applications* 8, 5 (1988), 41–49. 2, 7

- [WF80] WAINER H., FRANCOLINI C. M.: An empirical inquiry concerning human understanding of two-variable color maps. *The American Statistician* 34, 2 (1980), 81–93. 8
- [WSR\*20] WARE C., SAMSEL F., ROGERS D. H., NAVRATIL P., MOHAMMED A.: Designing Pairs of Colormaps for Visualizing Bivariate Scalar Fields. In *EuroVis 2020 - Short Papers* (2020), Kerren A., Garth C., Marai G. E., (Eds.), The Eurographics Association. doi: [10.2312/evs.20201047](https://doi.org/10.2312/evs.20201047). 8
- [WTB\*18] WARE C., TURTON T. L., BUJACK R., SAMSEL F., SHRI-VASTAVA P., ROGERS D. H.: Measuring and modeling the feature detection threshold functions of colormaps. *IEEE transactions on visualization and computer graphics* 25, 9 (2018), 2777–2790. 3
- [WTS\*17] WARE C., TURTON T. L., SAMSEL F., BUJACK R., ROGERS D. H.: Evaluating the Perceptual Uniformity of Color Sequences for Feature Discrimination. In *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)* (2017), Lawonn K., Smit N., Cunningham D., (Eds.), The Eurographics Association, pp. 7–11. doi: [10.2312/eurorv3.20171107](https://doi.org/10.2312/eurorv3.20171107). 2
- [WVVWVDL08] WIJFFELAARS M., VLIEGEN R., VAN WIJK J. J., VAN DER LINDEN E.-J.: Generating color palettes using intuitive parameters. *Computer Graphics Forum* 27, 3 (2008), 743–750. URL: <http://dx.doi.org/10.1111/j.1467-8659.2008.01203.x>, doi: [10.1111/j.1467-8659.2008.01203.x](https://doi.org/10.1111/j.1467-8659.2008.01203.x). 6
- [ZH16] ZHOU L., HANSEN C.: A survey of colormaps in visualization. *IEEE Transactions on Visualization and Computer Graphics* 22, 8 (2016), 2051–2069. 2, 6
- [ZM06] ZHANG H., MONTAG E.: Perceptual color scales for univariate and bivariate data display. *The Society for Imaging Science and Technology (IS&T)* (2006). 6
- [ZSN20] ZELLER S., SAMSEL F., NAVRATIL P.: Environmental visualization: Moving beyond the rainbows. In *Practice and Experience in Advanced Research Computing*. 2020, pp. 321–326. 8
- [ZWZ\*19] ZENG Q., WANG Y., ZHANG J., ZHANG W., TU C., VOILA I., WANG Y.: Data-driven colormap optimization for 2d scalar field visualization. In *IEEE VIS* (2019). 3