EG UK Computer Graphics & Visual Computing (2017)
Tao Ruan Wan and Franck Vidal (Editors)

Data Painter: A Tool for Colormap Interaction

O. H. Nagoor[1] R. Borgo[2] M. W. Jones[1]

[1] Swansea University, Department of Computer Science, Visual Computing
[2] King's College London

Abstract

The choice of a mapping from data to color should involve careful consideration in order to maximize the user understanding of the underlying data. It is desirable for features within the data to be visually separable and identifiable. Current practice involves selecting a mapping from predefined colormaps or coding specific colormaps using software such as MATLAB. The purposes of this paper are to introduce interactive operations for colormaps that enable users to create more visually distinguishable pixel based visualizations, and to describe our tool, Data Painter, that provides a fast, easy to use framework for defining these color mappings. We demonstrate the use of the tool to create colormaps for various application areas and compare to existing color mapping methods. We present a new objective measure to evaluate their efficacy.

CCS Concepts

•***Human-centered computing*** → *Scientific visualization; Visual analytics; Visualization toolkits;*

1. Introduction and Motivation

Colormapping is an important aspect of visualization. A transformation is created that converts each source data value into a destination pixel color. The goal is to create an image that effectively communicates those features of the data that are most salient to the task at hand. Typical tasks are: finding the absolute value, relating values across the image (e.g. temperature or wind speed across a weather map), identifying features or structure (e.g. searching for faulty electrical cables in thermal images). There may be several possibly conflicting requirements for understanding colormapped images, including: preserving data gradients or smoothness; features should be visually separable and identifiable; the color image should introduce perceptually higher dynamic range compared to using a gray scale representation (or equivalent).

The colormapping should be presented to the user as a labeled color scale. This enables a reading of the data where judgment may be made about absolute or relative values. Stability of colormap choice and familiarity with a color scale for a particular task can increase performance in identifying or reading the underlying data (e.g. a weather service utilizing a widely established temperature colormap where blue is cold and red is hot). In such situations the user is less dependent on making reference to the color scale in order to make relative data value judgments. Colormap design has focused on selection of colors to give a perceptual monotonic increasing color scale and research has suggested that linearly perceptual colormaps aid relative judgments.

Cartography is an example of an area where well-defined colormaps have been developed, e.g. for topographical relief shading. These are now so familiar that they serve to both produce visually appealing imagery for communication, and useful visualizations for scientific purposes. Tools and approaches exist that allow users to interact with their images using colormaps. Some of these can introduce problems. For example, pre-defined color look up tables with typically 256 colors, map the original data source data to a lower dynamic range. This can be countered by using control points that define a colormap which can be used within code with interpolation to create the colormap to any dynamic range. Environments (such as MATLAB, ParaView) provide colormaps that can be utilized within user code. We cover some of these approaches in section 2. Our contribution is a new interaction technique with colormaps and an objective evaluation function. Section 3 presents the interaction model and principles for interacting with colormaps. Section 4 describes and demonstrates our tools. In Section 5 we develop an objective function for evaluation purposes. The remainder of the paper provides case studies and discussions.

2. Existing Literature

Zhou and Hansen [ZH16] present a comprehensive survey of colormaps used in the field of visualization. The review classifies the methods of generating colormaps into four main categories, which are procedural methods, user-study based methods, rule-based methods and data-driven methods. The work contribution forms a perfect reference for researchers in terms of understanding the foundation of color spaces and color appearance models as well as it assists in determining the appropriate colormap for various data, tasks and applications. [SSM11] introduce a colormap-

DOI: 10.2312/cgvc.20171280

ping review with a number of guidelines and suggested tools to help not only visualization experts but also non-expert users in the task of finding the convenient color scales for their data or tasks.

Selecting an informative colormap is challenging even through the existence of the standard colormaps in visualization. There are many factors involved in making the decision of which colormap to be used including the data types, tasks, and consideration of personal perceptual differences (i.e. color vision deficiency (CVD)) [Oli13, NB13]. Some of the well-known colormaps are classified under the following types: sequential, binary, qualitative, and diverging [Bre94]. Each colormap performs well depending on the given goals and the nature of datasets [War88, Mor09]. Rainbow, diverging, grey-scale, heated body, categorical and isoluminant are examples of common colormaps used widely in visualization [ZH16, Kov15].

Samsel et al. [SPG*15] implement a nested colormapping technique in order to effectively maximize the perceptual range of an Ocean dataset. Using such a colormap significantly emphasizes the surface temperature in high-resolution data. Moreover, it increases the comprehension of the structure details within the area of interest than that of the standard colormaps. The approach only allows one additional colormap, and does not provide a user friendly (non-coding) approach. A version interpolating multiple colormaps was introduced by [SKP*16]. This interactive tool allows building customized colormap, which causes improvement in the perceptual reach of the data. However, the tool was only applied on the Ocean dataset, while our work demonstrates the effect on various narrow data range such as thermal images. Moreover, the dataset they used has a clear delineation in scale where the land is all warmer than the ocean, which causes the result to be segmented into land and sea. Usually, data will not contain such well-defined edges as we see with the Jupiter data set or other examples shown in this paper. Such case would lead to visually undesirable artifacts (although these are correct) when mapped with complex colormaps. We solve these problems by providing filtering techniques discussed in section 4. We also include an evaluation procedure, which was measured using an objective function. We also introduce image space filtering (lens view), and data space filtering (histogram selection), which further enhance the flexibility and the learning curve of our interactive tool. Another novel method for generating colormaps was implemented by Waldin et al. [WBRV16] which relies on measuring the user perceptual abilities to create an appropriate colormap specific for color vision deficiencies. The generated colormap is adapted though a game in which the user is required to arrange the colors from 3D color space into a line of smooth gradient. In Kindlmann et al. [KRC02], the user is required to select from double face images with luminance variations in order to measure their sensitivity to the light. The evaluation includes a repetitive procedure and each time with different color. Depending on the user selections, a new isoluminant colormap will be formed. Thompson et al. [TBSP13] use a random sampling procedure and CDF-based interpolation scheme. Although their algorithm requires a data analysis stage, it is scalable for large datasets since the sampling size is independent of the data size. In addition, the designed colormap highlight more features of the dataset as opposed to the ones colored by standard colormaps. Optimizing generated colormaps was presented by Fang et al. [FWD*17].

Transfer functions (TF) are an essential visualization tool used for mapping volumetric data not only to colors but also to opacity. Traditionally transfer functions design can be classified as being either data driven, where optical properties are assigned to elements in the dataset, or image driven, where transfer functions parameters are derived from analysis of rendered images.

Data driven approaches instead often make use of histogram to aid the analysis of a dataset scalar field and determine its properties. Histograms represent graphically the distribution of values along the different dimensions. Maciejewski et al. [MJW*13] showed how overall, when presented with the extra information carried by the histogram, both novice and expert users were able to provide better renderings due to the abstracted histograms helping to target, search and analyse the volume dataset. Correa and Ma [CM11] introduced the idea of visibility histograms, which represent the contribution that a given voxel has in the final rendering. Visibility histograms represent the distribution of a given visibility function in relation to the domain values of the volume. The computed visibility histogram can then be used to help discover occlusion patterns in the data. Ruiz et al. [RBB*11] improved on Correa's work by generating a transfer function automatically by minimizing the Kullback–Leibler divergence between the observed visibility distribution and a target visibility distribution. Wang et al. [WZC*11] extended the idea of visibility histogram to feature visibility where the opacity of each feature is generated automatically based on a user-defined visibility value.

Cai et al. [CTN*13] proposed a system for automatic transfer function design based on visibility distribution and projective colormapping. A preprocessing module first computes a 2D histogram, which is defined by the intensity and gradient magnitude information. The opacity transfer function is automatically derived by matching the observed visibility distribution to a target visibility distribution, which automatically set based on the IGM histogram of the volume. To compute the colour mapping, the histogram is segmented into several regions, the center of each region is then mapped to the CIELAB color space in order to assign a unique color for each region.

Ljung et al. [LKG*16] provide an inclusive review, which covers TF concepts, classifications, designing approaches and some valuable contributions. The classification covers a wide range of topics related to TFs such as dimensionality, derived attributes, aggregated attributes, rendering aspects, automation, and user interfaces.

While analytical techniques can help in determining data characteristics such as intensity, frequency and distribution, choosing the most appropriate colormaps is not an easy task: colors are not equally distinguishable by observers and are influenced by both task and application domain. Healey [Hea96] developed a procedure for designing sets of easily distinguishable colors [25] while Bergman et al. [BRT95] have extended his idea creating a taxonomy based on principles of perception, visualization tasks, and data types. Bergman et al.'s taxonomy provides a tool to develop and choose effective color scales for specific data types and goals.

Design of an appropriate colormap should be influenced not only by the data that is being visualised but also by the experience of the user to which the visualisation is aimed. Domain experts have the capability to support and inform the design process while novice

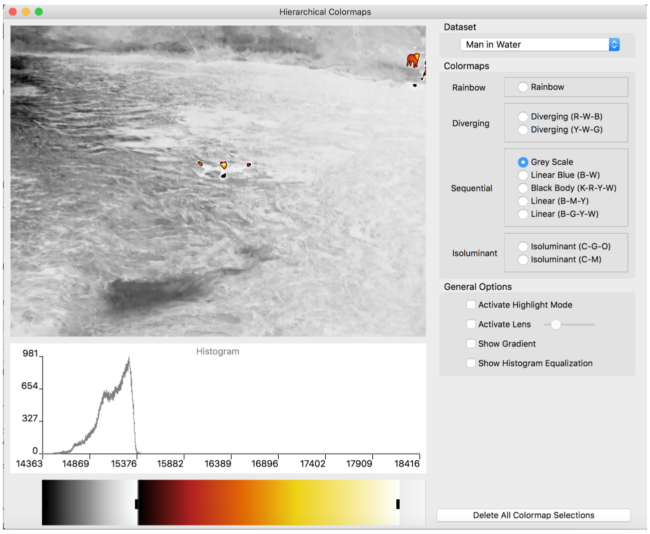

Figure 1: *The main interface, showing the image derived from the colormap, the histogram of the data, and current user colormap. The currently selected colormap can be swapped using radio buttons on the right.*

users might need different types of support to guide their own design. Borland and Huber [BH11] presents guidelines derived from close collaboration between visual designers and domain expert to create effective colormaps.

Akyüz and Kaya [AK16] create a metric and user study for (High Dynamic Range) HDR false colormaps. They study the data preparation stage, where various HDR techniques are employed, with Histogram Equalization featuring well in each study. They compare various colormaps in the user study and propose a metric based on CIE color differences [SWD05] that we augment in Section 5. Dynamic reduction can lead to more visually pleasing images, but if the colormap has a linearly perceptual scale, then this relationship is lost during the data preparation. High dynamic range reduction is not a feature of our work, although any data preparation stages may be integrated with our approach.

There is currently a lot of work involving coding colormaps for visualization; single linear color, diverging [Mor09], and multi-color and additional features such as isoluminant or linearly perceptually varying colormaps [Kov15]. Our work is colormap agnostic. We seek to avoid the need for users to resort to coding to obtain the required effects. We propose a user interface for interactive colormap placement. The above effects (such as nested colormaps) can be set up in seconds with our proposed interface. We also propose an objective function for evaluating the results of the interaction, i.e., the efficacy of the overall resultant colormap.

3. Design

In addition to being informed by the above literature, we are also influenced by the design principles from working with time-series data [WBJ16, WJL*15]. If we regard that there are similarities between interacting with the histogram of the source data range and a time-series data set, then the functionality of working with time-series data such as those provided by Chronolens [ZCPB11] and SignalLens [Kin10] readily transfer to this area. Close collaborations with domain experts has also reiterated how interaction with a visualization tool can strongly influence understanding of the data as well as understanding and effectiveness of the tool itself [TM04].

Based on state of the art literature and personal experience, we identified a number of general design principles applicable for producing colormapped images from source data:

P1 Multiple linked views. The user interface should link the view of the resultant colormapped image with the currently user defined colormap and the histogram of the data (Figure 1).

P2 Responsive views. These views should be responsive to user interaction. During interaction with the colormap, feedback on the interaction and the resultant colormapped image should be updated dynamically.

P3 Direct manipulation. The interface should allow direct manipulation [Shn83] of the current hierarchical colormap using an intuitive mouse control. The user should be able to modify the data range over which a colormap spans. This can be achieved by allowing a fixed range but moving the colormap (translation), or allowing the range to change (resizing). The selected colormap should be swappable for an alternative. Colormaps can be introduced to and deleted from the current range.

P4 Histogram and histogram equalization view. Introducing a histogram of the data allows the user to identify peaks in the data. The user should be able to zoom to regions of interest in the data range, and thus focus new colormaps in that area to improve the visual dynamic range. The same plotting area should illustrate the histogram equalization of the data when the user activate it.

P5 Interactive filtering. The system provides two filtering methods in data space and in image space. Both filtering should be user-friendly and improve the perceptual outcome of the data.

P6 Swappable colormaps. The system should allow a wide variety of colormaps that the user can easily swap between. Essentially our choice for inclusion was steered by readily available colormaps (lists of control points) that we could utilize in our code. Colormaps additional to the ones available could be introduced. As we started with thermal images, we selected blackbody as the initial default colormap.

P7 Objective evaluation. We provide statistics about the current colormapping. These were introduced for the objective evaluation, but they may also aid the user in determination of the colormapping.

Our design principles are not bounded to the development of a single tool but rather aim at capturing those significant qualities, at both interaction, data processing and image generation levels, that make the colormap production pipeline most successful.

4. Implementation details

To provide the bridge between the intuitive direct control via mouse input and the necessary functionality (**P3**), we view the complete resultant colormap as a back to front hierarchy of individual colormaps. Each colormap has left and right attributes determining

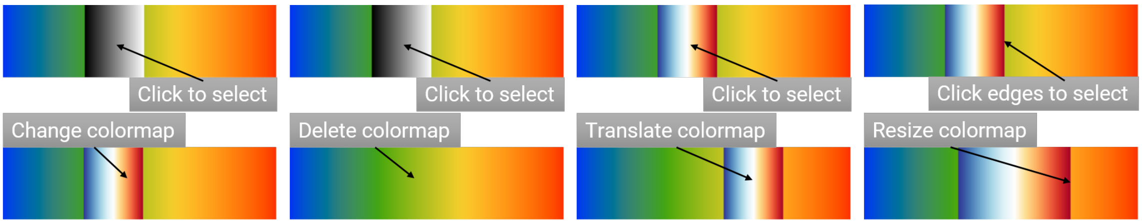

Figure 2: *Direct manipulation of a selected colorpmap (left button): Left to right, changing the selected colormap to a new one, deleting the selected colormap, translating the colormap and resizing the colormap by moving either edge.*

where in the data range it starts and finishes. An initial background colormap covers the whole data range. Each colormap also has an id number and a colormap number (which specific colormap it is currently instantiated as (**P6**)). All user interaction is converted into actions on these attributes. Whenever any attribute is changed, the overall colormapping is evaluated in order to produce the colormapped image, which is dynamically refreshed during interaction (**P1, P2**).

P3 Direct manipulation. In the following, we differentiate between whether the user is selecting an existing highlight to work with, introducing a new one or interacting with the overall colormap widget.

Selected colormap (P3, P6). If a colormap in the current hierarchy has focus, the direct manipulation operations alter the attributes for that colormap and refreshes the display. The included operations are depicted in Figure 2. If a colormap is selected, the user can use the radio buttons to select a different colormap, Figure 2 (left). The user may delete the currently selected colormap, Figure 2 (inner left). By clicking to select the colormap, then holding the mouse button and moving the mouse (mousemove) the user can directly translate the colormap along the range of the data, Figure 2 (inner right). By clicking to select one or other of the edges of the colormap, then holding the mouse button and moving (mousemove) the user can change the position of that edge in the range of the data, which effectively resizes the colormap, Figure 2 (right). Dynamic feedback is provided at each event. (**P1, P2**)

New colormap. If there is no currently selected colormap, the mouse click is interpreted as the creation of a new colormap (**P3**). The user holds the mouse button down, and moves to create a colormap of the desired size covering the required data range. This new colormap is placed at the front of the hierarchy with the desired data range, and uses the current colormap type determined from the radio buttons. Dynamic feedback is provided as the user resizes the new colormap. (**P1, P2**)

Colormap widget. The overall colormap is also treated as a component (widget) that can be manipulated directly. The user may zoom or translate the widget. After or during any number of transforms, the user is able to employ any of the selected colormap or direct manipulation techniques above to alter existing or create new colormaps within the hierarchy. Figure 3 illustrates a sequence of interactions to demonstrate the system functionality. The user starts from the default colormap and introduces a new grayscale colormap. The colormap widget is zoomed and translated. The grayscale colormap is resized to a very small range of interest in the original data source. An additional divergent col-

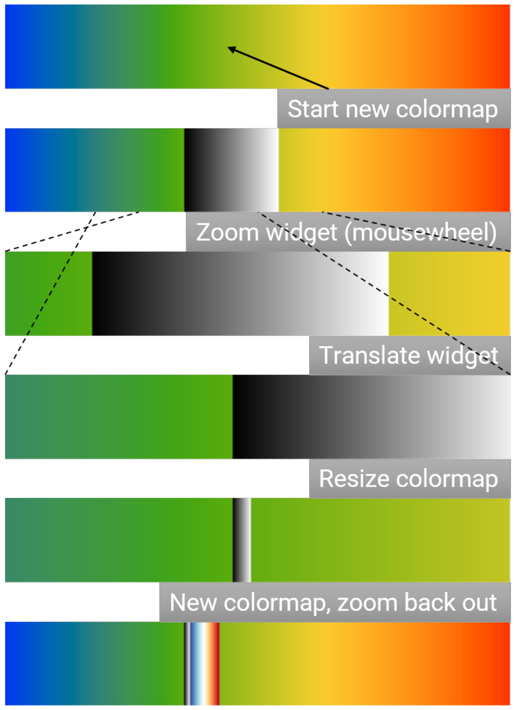

Figure 3: *Interaction with the colormap widget. The user introduces a new colormap, zooms and translates the widget and then resizes that colormap to occupy a very small range of the data. After zooming out, this small range is seen to a the left of another introduced colormap. Black dotted lines show the subinterval for the zoom level before and after translation.*

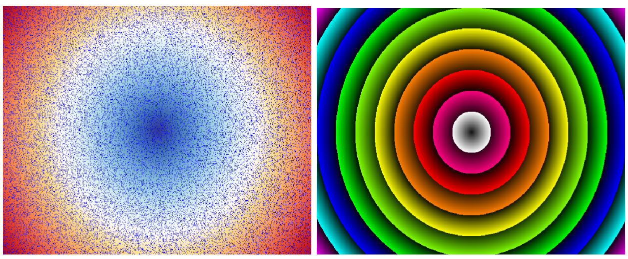

Figure 4: *Distance field mapped using divergent colormap (left) and multiple linear colormap (right) from Table 1* (512×512)

colormap	MSE Gradient	Angle	$\Delta E_{CIE00} > 1$
	0.04	70.0%	97.9%
	0.08	37.8%	96.7%
	0.02	80.8%	98.2%
	0.06	69.0%	98.0%
	0.001	100.0%	99.6%

Table 1: *Distance Field (Figure 4). For each corresponding colormap, gradient indicates the deviation from the true gradient ($\nabla = 1$) for distance field, smaller is better), and angle indicates the percentage of vectors that are within $\pm 10^o$ of the ground truth (larger is better). The percentage of pixel pairs satisfying $\Delta E_{CIE00} > 1$ is in the last column.*

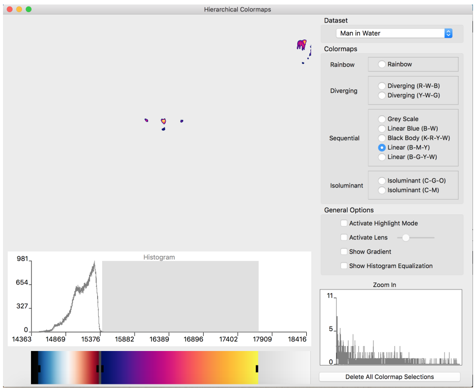

Figure 5: *Illustrating the effect of filtering in data space by selecting a range of values in the histogram view. Also, a zooming widget will appear in the right-bottom corner to show the selected range.*

ormap is placed against the grayscale, and then the user zooms out to the original overall colormap view.

Histogram and histogram equalization widget. (P4) We keep the histogram widget updated with the same view transform as the colormap widget. The user can use the peaks in the histogram to quickly locate colormaps for highlighting that data. We enable the ability to pan and zoom the histogram because in the case of thermal images there is a wide range of data with high numbers of samples in narrow thermal ranges of interest, and also long tails of data values outside those ranges. The user is able to zoom right into the histogram, and have very precise control over the placement of a colormap over the range of data of interest, or can sweep a colormap back and forth over the histogram, to explore and highlight different sample peaks as it is illustrated in Figure 5.

Interactive filtering. (P5) We present two filtering methods that further improve the flexibility and the perception of the underlying data without any prior knowledge of its distribution. The first filtering procedure is in data space filtering. The user is allowed to select a proportion of the data by selecting part of the data histogram. With such selection, the user can identify where data values are located and what they represent such as the example illustrated in figure 5 (the human body and the dog). The second filtering feature is in the image space, where the customized colormap will be applied on part of the rendering result (lens view) illustrated in figure 6. The size of the lens can easily be adjusted to the desired size. Such method allows the user to distinguish the narrow ranges with high depth perception as well as to compare the rendered result with and without the effect of the dense colormap. Adding these two features help in building the appropriate colormap. Moreover, it helps in gaining more knowledge about the dataset as well as optimizing the learning process.

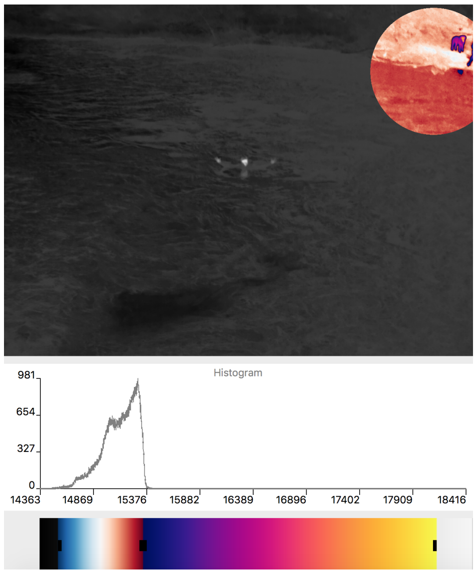

Figure 6: *Illustrating the effect of filtering in image space. The result of the customized colormap is shown within an adjustable lens.*

Swappable colormaps. (P6) The overall system may be seen in Figure 1. The colormapped image of the data source is seen at the top. Below this are the histogram and colormap hierarchy widgets. The user directly interacts with these components as above. On the right hand side we currently have a small selection of colormaps categorized according to the various types [Kov15]. The user can select any of the colormaps using the radio buttons. If a colormap is selected in the hierarchy, then clicking on a different radio button changes the type. If nothing is selected, then the radio button indicates the type of colormap that will be created if the user introduces a new colormap to the hierarchy.

Objective evaluation. (P7) To evaluate the discriminating power of user controlled hierarchical colormaps we develop an objective function in Section 5. This has a slight impact on the speed of interaction. During interaction with the colormaps, it is interesting to get dynamic feedback from these measures, and to manipulate the colormaps to optimize the values. It would require further study (a formal user study) to determine if by using these measures the user is able to effectively and usefully increase the discriminating power of the colormaps from a user perspective.

5. Objective Evaluation

In this section, we construct an objective function for measuring the effectiveness of a colormapping. First, we model the following constraints: data that is locally smooth in the source data set should result in a locally smooth area in the colormapped image. The color image should attempt to minimize or enhance the dynamic range from the source data. The former constraint penalizes colormaps that introduce new features or noise to the image. The latter con-

colormap	MSE Gradient	Angle	$\Delta E_{CIE00} > 1$
	3.3	85.7%	85.8%
	10.9	71.8%	71.7%
	1.6	92.0%	91.9%
	1.9	90.6%	94.8%
	0.88	95.19%	93.9%

Table 2: *Man in water (Figure 7). For each corresponding colormap, MSE is the MSE of the estimated gradient from the colormap to the gradient estimated from the data, smaller is better), and angle indicates the percentage of vectors that are within $\pm 10^o$ of the estimated gradient from the original data (larger is better). The percentage of pixel pairs satisfying $\Delta E_{CIE00} > 1$ is in the last column.*

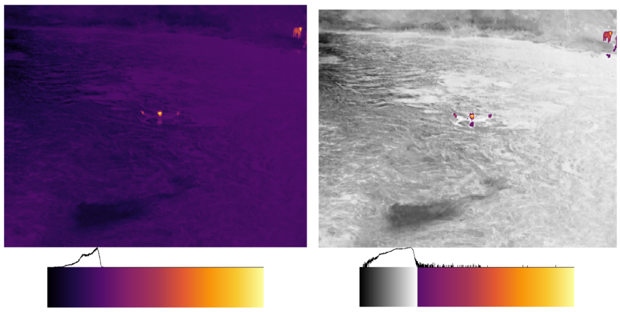

Figure 7: *Thermal image data (640×480): (Left) Man in water with Inferno colormap. Note the histogram shows much of the data (water) towards the left. (Right) The user has replaced the lower data with a grayscale colormap giving more detail in the water. The logarithmic histogram helped to determine that higher values were present across the data range.*

straint penalizes colormaps that reduce the dynamic range of the data relative to others. We also model a constraint that the direction of the derivative of the colormapped image should be consistent with that of the source data, also to avoid the introduction of features into the image that are not present in the data.

Given the data source $f_D(x,y)$, the gradient is $\nabla f_D(x,y) = \left(f'_x = \frac{\partial f_D}{\partial x}, f'_y = \frac{\partial f_D}{\partial y}\right)$, and the gradient magnitude is $|\nabla f_D(x,y)| = \sqrt{f'^2_x + f'^2_y}$, where, using central differences, $\frac{\partial f_D}{\partial x} = \frac{f_D(x+d) - f_D(x-d)}{2d}$. We derive similarly, for the colormapped image $f_C(x,y)$.

Two alternate colormappings C_1 and C_2 can be compared, such that C_1 fulfills the requirements better than C_2 if: (i) MSE(C_1)<MSE(C_2) (MSE(C)=$\frac{1}{n}\sum(|\nabla f_C(x,y)| - |\nabla f_D(x,y)|)^2$, the Mean Squared Error of C_1 is less than C_2). (ii) $\hat{\nabla f_D}(x,y) \cdot \hat{\nabla f_{C_1}}(x,y) < \hat{\nabla f_D}(x,y) \cdot \hat{\nabla f_{C_2}}(x,y)$ (the gradient direction produced by C_1 is closer to the original data gradient). d is the interval chosen for central differences. Larger d weakens the locality constraint, therefore we use $d = 1$ (pixel neighbors).

Akyüz and Kaya [AK16] propose that the best mapping for false color images should have the highest perceivable color differences between pixel values measured in CIEDE2000 color differences (they denote ΔE_{CIE00}). They measure the number of pixel pairs where $\Delta E_{CIE00} > 1$ and rank algorithms accordingly on a test suite. This metric favors colormaps with varying colors which may introduce perceptual features in the image that are not present in the data. The work is supported by a psychophysical study in which stimuli are pairs of patches from images, and participants must determine which of the pair has the highest luminance.

The CIEDE2000 metric [AK16] favors images with high color differences, which may not be representative of the original data set. Therefore we propose that a good objective measure can be made by combining the goals of matching the direction of the original data, and enhancing or maintaining the gradient as closely as possible in a local area, along with a global goal of color differences measured as above using ΔE_{CIE00}. This is supported by the user study of Akyüz and Kaya [AK16], and our goals about accuracy and enhanced perception from source data.

We demonstrate this with a case study of a distance field generated from a point source at the origin, O, such that $f_D(x,y) = |(x,y)|$. Since it is generated under the Euclidean distance norm, $|\nabla f_D(x,y)| = 1$ and $\nabla f_D(x,y) = (x,y)$ is the ground truth. We create a 256^3 entry vector, V, and as each RGB color triple is generated during mapping, we set the corresponding position ($pos = r << 16|g << 8|b$) in this vector to be the minimum value that generated that color so far ($V[pos] = v$). This provides a fast LUT to obtain data value v, from color, which simulates the viewer's interpretation of value from color. Even for double precision data, this is only 128MB.

For a set of 50,000 random points without replacement the gradient direction and magnitude for different colormaps is compared against the ground truth. The colormaps used are black body, gray, divergent colormap from Color Brewer [Bre94], rainbow_bgyr_35-85_c72_n256 [Kov15] and a colormap comprising 10 contiguous single color linear colormaps. Figure 4 (left) shows the divergent colormap test with the test points colored blue. Table 1 presents the results with column $\Delta E_{CIE00} > 1$ giving the percentage of pixel pairs that satisfy that test (40,000 pairs randomly chosen from the test image).

Of the colormaps used, the one with the least colors is the grayscale (two fixed control points - start and end). Unsurprisingly it produces images which least discriminate the data. The real valued data is quantized into 256 different (gray) colors. This is shown in Table 1. Only 38% of the gradients derived from the grayscale colormapped image are within $\pm 10^o$ when compared to the ground truth gradient. The $\Delta E_{CIE00} > 1$ test also has lowest value for the grayscale, but we observe that this measure is not so discriminating. The final color scale was chosen because it is very suitable for visualizing distances (Figure 4 (right)), and provides lower degree of quantization compared to the other colormaps. Correspondingly the derived gradient is more accurate, achieving 100% of gradients within $\pm 10^o$ of the ground truth. If we increase the CIE delta test to be $\Delta E_{CIE00} > 3$ the final column becomes slightly more discriminating at respectively 93.8%, 90.1%, 94.6%, 93.6% and 98.7%.

In the case of the distance field, we have the analytical ground truth (x,y). For general data sources, we do not have access to the analytical ground truth. To obtain a ground truth for those data sources, we compute the gradient using central differences. In the case of the distance field, we computed the objective measure in

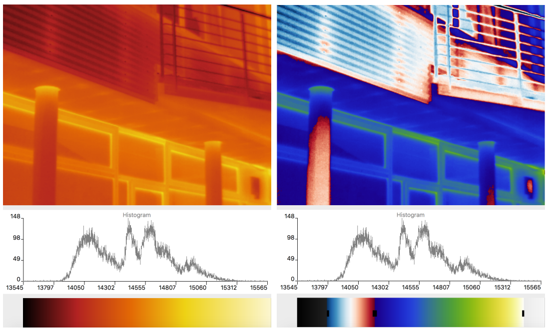

Figure 8: *Building images (Left) Original data. (Right) the user customized colormap*

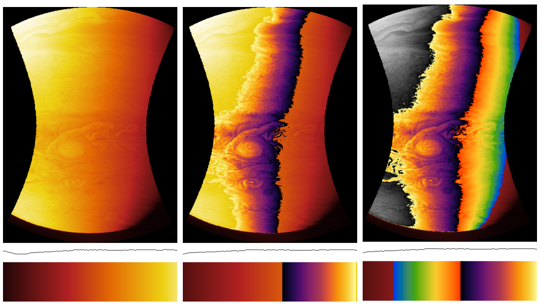

Figure 9: *Juno spacecraft imagery* (1600×1600)*: (Left) Original data. (Middle) The user has introduced the inferno colormap to highlight the great red spot. (Right) Other colormaps are introduced to increase the visible dynamic range of otherwise saturated features (The rightmost colormap has been zoomed and translated).*

two situations. The first used the analytical gradient ground truth. The second used the gradient calculated using central differences from the source data. There was very little variation in the results. Therefore, we conclude, that for data sets where the precise ground truth gradient is unknown, the gradient calculated according to central differences from the original data set is a good surrogate.

We also perform the same experiment on the Man in Water thermal image which demonstrates similar results (Table 2). This time we replace the last colormap with a user defined colormap from our tool where the water has been set to grayscale, and the person in the water (and dog and second person's leg) are set to the rainbow_bgyr_35-85_c72_n256 colormap. The user defined colormap has the lowest MSE on the gradient, and the highest agreement between colormap derived gradient ($\nabla f_C(x,y)$) and data gradient ($\nabla f_D(x,y)$) which is an indication of the usefulness of our approach.

6. Case Studies

Thermal image data has a larger dynamic range. They are becoming more useful as a diagnostic tool for building issues, e.g., detecting faulty cables (hot), water leaks (cool), problems with air permeability or other insulation issues. In these cases fault detection requires good contrast to surrounding background temperature (cables, water), or good detection of gradients (insulation issues).

Man in water (figure 7). First we include one typical example of using thermal images to detect people against the cooler background. The original data with a default inferno colormap produces the best highlighting of the person, but the details in the water (which are not really relevant to the task) are largely undefined. From the histogram in (figure 7 (left)), we can see that most of the data values are to the darker part of the colormap, and because there are so few samples at the higher thermal values, these are lost in the histogram. Switching to the log histogram allows us to see those sample values are spread higher through the histogram. With a few seconds of interaction, the user is able to place a grayscale colormap which better highlights the water, where we can now see the details (if these were indeed interesting). The hot objects remain on the original inferno colormap where they are now more visible. Table 2 presents the results of running the objective evaluation on the image. The user defined colormap produces a more accurate representation of the gradients in the image.

Building (figure 8). For the blackbody colormap, the image produces $\Delta E_{CIE00} > 1 = 90.1\%$, MSE$= 0.92$ and gradients within $\pm 10^o = 88.9\%$. For the user colormap, the image produces $\Delta E_{CIE00} > 1 = 95.8\%$, MSE$= 0.28$ and gradients within $\pm 10^o = 95.5\%$ which is better in every measure. This is also visible in the image. The user has created a false color image using multiple colormaps. Each has been positioned to highlight a different color range, and in this case it visibly separates building materials. Within each material, better gradients are observable (which is also confirmed by the gradient measures). The image has a larger distinguishable color range (also confirmed by the ΔCIE measure). Heat loss temperature gradients are now visible across window panes and joists in the ceiling, and could be used to inform further insulation decisions.

Jupiter. The Jupiter image (Figure 9) presents a different challenge. The original data has a high variation in light towards the sun. Large areas of the image have low gradient magnitude on the blackbody colormap. By creating a more focused colormap over a smaller range of data, the user can sweep this back and forth across the data range, using it like a data lens across the image. The middle image shows the inferno colormap being swept across, and highlighting the great red spot. In the right image, the user has introduced further colormaps to enhance the difficult to distinguish areas.

7. Conclusions

Colormaps are used ubiquitously to provide a scale from which the viewer is able to deduce certain information about absolute or relative data values from the colored data. Previous work has studied the importance of perception from colormaps with regard preserving linear relationships within the data. High dynamic range data provides challenges, since the colormapping reduces the original dynamic range. Usual measures involve transforming the data. Previous work [AK16] studies several approaches for transforming HDR images, along with false color coding. In this approach we retain the original source data but transform the colormap with user interaction. Our tool allows the interactive placement and transformations of colormaps within the data range with real-time visual feedback. The user is able to sweep colormaps across the image in order to detect features of interest. The resulting overall colormap is regarded as a front to back hierarchy of individual colormaps. The method is shown to have higher dynamic range than standard colormap images and is able to highlight more features and structural details.

References

[AK16] Akyüz A. O., Kaya O.: A proposed methodology for evaluating hdr false color maps. *ACM Trans. Appl. Percept. 14*, 1 (July 2016), 2:1–2:18. doi:10.1145/2911986. 3, 6, 7

[BH11] Borland D., Huber A.: Collaboration-specific color-map design. *IEEE Computer Graphics and Applications 31*, 4 (July 2011), 7–11. doi:10.1109/MCG.2011.55. 3

[Bre94] Brewer C. A.: Color use guidelines for mapping and visualization. *Visualization in modern cartography* (1994), 123–148. 2, 6

[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* (Washington, DC, USA, 1995), VIS '95, IEEE Computer Society, pp. 118–. 2

[CM11] Correa C. D., Ma K.-L.: Visibility histograms and visibility-driven transfer functions. *IEEE Transactions on Visualization and Computer Graphics 17*, 2 (Feb. 2011), 192–204. doi:10.1109/TVCG.2010.35. 2

[CTN*13] Cai L., Tay W.-L., Nguyen B. P., Chui C.-K., Ong S.-H.: Automatic transfer function design for medical visualization using visibility distributions and projective color mapping. *Computerized Medical Imaging and Graphics 37*, 7–8 (2013), 450 – 458. doi:10.1016/j.compmedimag.2013.08.008. 2

[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. 2

[Hea96] Healey C. G.: Choosing effective colours for data visualization. In *Visualization '96. Proceedings.* (Oct 1996), pp. 263–270. doi:10.1109/VISUAL.1996.568118. 2

[Kin10] Kincaid R.: Signallens: Focus+context applied to electronic time series. *Visualization and Computer Graphics, IEEE Transactions on 16*, 6 (Nov 2010), 900–907. doi:10.1109/TVCG.2010.193. 3

[Kov15] Kovesi P.: Good colour maps: How to design them. *CoRR abs/1509.03700* (2015). 2, 3, 5, 6

[KRC02] Kindlmann G., Reinhard E., Creem S.: Face-based luminance matching for perceptual colormap generation. In *IEEE Visualization, 2002. VIS 2002.* (Nov 2002), pp. 299–306. doi:10.1109/VISUAL.2002.1183788. 2

[LKG*16] Ljung P., Krüger J., Groller E., Hadwiger M., Hansen C. D., Ynnerman A.: State of the art in transfer functions for direct volume rendering. *Computer Graphics Forum 35*, 3 (2016), 669–691. doi:10.1111/cgf.12934. 2

[MJW*13] Maciejewski R., Jang Y., Woo I., Janicke H., Gaither K. P., Ebert D. S.: Abstracting attribute space for transfer function exploration and design. *IEEE Transactions on Visualization and Computer Graphics 19*, 1 (Jan. 2013), 94–107. doi:10.1109/TVCG.2012.105. 2

[Mor09] Moreland K.: *Diverging Color Maps for Scientific Visualization.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 92–103. doi:10.1007/978-3-642-10520-3_9. 2, 3

[NB13] Nigam P. K., Bhattacharya M.: Colour vision deficiency correction in image processing. In *2013 IEEE International Conference on Bioinformatics and Biomedicine* (Dec 2013), pp. 79–79. doi:10.1109/BIBM.2013.6732581. 2

[Oli13] Oliveira M. M.: Towards more accessible visualizations for color-vision-deficient individuals. *Computing in Science & Engineering 15*, 5 (2013), 80–87. doi:10.1109/MCSE.2013.113. 2

[RBB*11] Ruiz M., Bardera A., Boada I., Viola I., Feixas M., Sbert M.: Automatic transfer functions based on informational divergence. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec 2011), 1932–1941. doi:10.1109/TVCG.2011.173. 2

[Shn83] Shneiderman B.: Direct manipulation: A step beyond programming languages. *Computer 16*, 8 (Aug 1983), 57–69. doi:10.1109/MC.1983.1654471. 3

[SKP*16] Samsel F., Klaassen S., Petersen M., Turton T. L., Abram G., Rogers D. H., Ahrens J.: Interactive colormapping: Enabling multiple data range and detailed views of ocean salinity. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI EA '16, ACM, pp. 700–709. doi:10.1145/2851581.2851587. 2

[SPG*15] Samsel F., Petersen M., Geld T., Abram G., Wendelberger J., Ahrens J.: 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* (New York, NY, USA, 2015), CHI EA '15, ACM, pp. 703–710. doi:10.1145/2702613.2702975. 2

[SSM11] Silva S., Santos B. S., Madeira J.: Using color in visualization: A survey. *Computers & Graphics 35*, 2 (2011), 320 – 333. Virtual Reality in BrazilVisual Computing in Biology and MedicineSemantic 3D media and contentCultural Heritage. doi:10.1016/j.cag.2010.11.015. 1

[SWD05] Sharma G., Wu W., Dalal E. N.: The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application 30*, 1 (2005), 21–30. doi:10.1002/col.20070. 3

[TBSP13] Thompson D., Bennett J., Seshadhri C., Pinar A.: A provably-robust sampling method for generating colormaps of large data. In *2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV)* (Oct 2013), pp. 77–84. doi:10.1109/LDAV.2013.6675161. 2

[TM04] Tory M., Moller T.: Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics 10*, 1 (Jan 2004), 72–84. doi:10.1109/TVCG.2004.1260759. 3

[War88] Ware C.: Color sequences for univariate maps: Theory, experiments and principles. *IEEE Computer Graphics and Applications 8*, 5 (1988), 41–49. doi:10.1109/38.7760. 2

[WBJ16] Walker J., Borgo R., Jones M. W.: Timenotes: A study on effective chart visualization and interaction techniques for time-series data. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of Information Visualization 2015) 23*, 1 (Janaury 2016), 549–558. doi:10.1109/TVCG.2015.2467751. 3

[WBRV16] Waldin N., Bernhard M., Rautek P., Viola I.: Personalized 2D color maps. *Computers & Graphics 59* (2016), 143 – 150. doi:10.1016/j.cag.2016.06.004. 2

[WJL*15] Walker J. S., Jones M. W., Laramee R. S., Bidder O. R., Williams H. J., Scott R., Shepard E. L. C., Wilson R. P.: Timeclassifier: a visual analytic system for the classification of multi-dimensional time series data. *The Visual Computer 31*, 6-8 (2015), 1067–1078. doi:10.1007/s00371-015-1112-0. 3

[WZC*11] Wang Y., Zhang J., Chen W., Zhang H., Chi X.: Efficient opacity specification based on feature visibilities in direct volume rendering. *Computer Graphics Forum 30*, 7 (2011), 2117–2126. doi:10.1111/j.1467-8659.2011.02045.x. 2

[ZCPB11] Zhao J., Chevalier F., Pietriga E., Balakrishnan R.: Exploratory Analysis of Time-series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Oct. 2011), 2422–2431. doi:10.1109/TVCG.2011.195. 3

[ZH16] Zhou L., Hansen C. D.: A survey of colormaps in visualization. *IEEE Transactions on Visualization and Computer Graphics 22*, 8 (Aug 2016), 2051–2069. doi:10.1109/TVCG.2015.2489649. 1, 2