Methods for Effective Color Encoding and the Compensation of Contrast Effects

Dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften

vorgelegt von

Sebastian Mittelstädt

an der

Mathematisch-Naturwissenschaftliche Sektion
Fachbereich Informatik und Informationswissenschaft

Tag der mündlichen Prüfung: 12. November 2015

1. Referent: Prof. Dr. Daniel A. Keim
2. Referent: Prof. Dr. Oliver Deussen
3. Referentin: Prof. Dr. Heidrun Schumann

Abtract

Color is one of the most effective visual variables to encode information. It is pre-attentively processed and encodes a variety of information such as categorical, ordinal, quantitative but also semantic information. However, the *effectiveness* of color encodings is not sufficiently defined and research proposes controversial guidelines. This thesis bridges the gap between the controversies by a novel definition of *effectiveness* and provides evidence that the *effectiveness* depends on the analysis task that is performed with color. Current guidelines provide effective color encodings for single elementary analysis tasks. However, for solving real world problems, in most practical applications, single elementary analysis tasks are not sufficient but need to be combined. This thesis proposes a set of novel quality metrics, design guidelines and methods to design effective color encodings for combined analysis tasks. First, for encoding single dimensions, and second, for high-dimensional data relations. For this purpose, the thesis provides novel tools that guide novice and expert designers through the creation of effective colormaps and allow the exploration of the design space of color encodings. The visualization expert is integrated in the design process to incorporate his/her design requirements, which may depend on the application, culture, and aesthetics.

Despite a well-designed colormap, optical illusions still bias the perception at the first level of the analysis process. For instance, in visualizations contrast effects let pixels appear brighter if surrounded by a darker area. This distorts the encoded metric quantity of the data points significantly, and even if the analyst is aware of these perceptual issues, the visual cognition system is not able to compensate for these effects accurately. To overcome these issues, this thesis presents the first methodology and the first methods to compensate for physiological biases such as contrast effects. The methodology is based on perceptual metrics and color perception models that can also be adapted to an individual target user. Experiments with over 40 participants reveal that the technique

doubles the accuracy of users comparing and reading color encoded data values. Further experiments show that the introduced personalized perception models significantly outperform existing perception models applied in contrast effect compensation. Thereby, this thesis provides a solution to the problem of contrasts effects in information visualization. However, the thesis also presents how contrast effects can be exploited and used to enhance visualizations. First, by boosting the visibility of important data points, or second, by increasing the readability of high-frequency visualizations such as network visualizations.

All methods, introduced in this thesis, can be used in any application or image without adapting to the visualization itself. Therefore, the *effectiveness* of the methods is demonstrated in use cases and case studies of different domains.

Zusammenfassung

Farbe ist eine der effektivsten visuellen Variablen, um Informationen zu kodieren. Sie wird präattentiv wahrgenommen und kann genutzt werden um kategorische, ordinale, quantitative sowie semantische Information darzustellen. Jedoch ist die Effektivität von Farbkodierungen nicht hinreichend definiert, was zu kontroversen Ergebnissen und Richtlinien für Farbkodierungen geführt hat. Um diese Lücke zu schließen, führt diese Dissertation eine neue Definition von Effektivität für Farbkodierungen ein und legt dar, dass die Effektivität einer Farbkodierung von der Analyseaufgabe abhängt, die mit Hilfe von Farbe gelöst werden soll. Mithilfe existierender Richtlinien können effektive Farbkodierungen für einzelne elementare Analyseaufgaben erstellt werden. Diese reichen jedoch nicht für reale Anwendungen aus und müssen kombiniert werden. Aus diesem Grund, stellt diese Dissertation neue Qualitätsmaße, Richtlinien und Methoden vor, um Farbkodierungen für kombinierte Analyseaufgaben zu erstellen. Einerseits für die Kodierung einzelner Dimensionen von Daten, aber auch für die Kodierung von hoch–dimensionalen Relationen von Daten. Diese Dissertation stellt Werkzeuge bereit, die Experten und Einsteiger durch die Erstellung effektiver Farbkodierungen leiten und damit die Exploration des Designraums für Farbkodierungen ermöglichen. Der Experte kann seine Erfahrungen einfließen lassen, um damit die Farbkodierung an die Anwendung, Kultur und Vorlieben seiner Nutzer anzupassen.

Egal wie effektiv eine Farbkodierung ist, optische Illusionen können den Nutzer negativ in seiner Analyse beeinflussen. Zum Beispiel bewirken Kontrasteffekte, dass Pixel auf dunklen Hintergründen heller wirken und auf hellen Hintergründen dunkler. Diese Effekte verzerren die Wahrnehmung der kodierten Daten signifikant. Selbst wenn der Analyst sich dieser Effekte bewusst ist, können sie nicht kognitiv kompensiert werden, da sie auf den unterbewussten Ebenen der Wahrnehmung geschehen. Aus diesem Grund stellt diese Dissertation die erste Methodologie und erste Methoden vor, um physiologische Illusionen wie Kontrasteffekte zu kompensieren. Die Methodologie nutzt Wahrnehmungs-

modelle und Metriken, die auf den individuellen Nutzer angepasst werden können. Durch Experimente mit über 40 Versuchspersonen kann gezeigt werden, dass die Methode die Genauigkeit von Menschen verdoppelt, die Werte mittels Farbe ablesen und vergleichen. Weitere Experimente legen dar, dass die eingeführte Personalisierung von Wahrnehmungsmodellen heutige Wahrnehmungsmodelle in der Kompensation von Kontrasteffekten signifikant verbessert.

Des Weiteren stellt diese Dissertation vor, wie Kontrasteffekte genutzt werden können, um Informationen zu kodieren und Visualisierung anzureichern. Einerseits um wichtige Informationen visuell hervorzuheben, andererseits um die Lesbarkeit von hoch-frequenten Visualisierungen wie Netzwerken zu verbessen.

Alle vorgestellten Methoden dieser Dissertation können auf jedes Bild und auf jede Visualisierung angewendet werden, ohne sie an die Visualisierungen anpassen zu müssen. Aus diesem Grund wird die Effektivität der Methoden an Beispielen und Fallstudien aus verschiedenen Domänen in dieser Dissertation demonstriert.

Danksagung

Diese Dissertation ist ein wichtiger Meilenstein meiner akademischen Karriere. Anfangs war es nicht leicht in diesem kontrovers diskutierten Gebiet Fuß zu fassen. Zum Glück fand ich die nötige Motivation und konnte schließlich die "Methoden zur Kompensation von Kontrasteffekten" veröffentlichen, welche den Kern dieser Dissertation bilden.

Aus diesem Grund möchte ich meinem Doktorvater Professor Dr. Daniel A. Keim danken, dass er, von Anfang an, an den Wert dieses wissenschaftlichen Beitrags geglaubt hat. Er hat mich nicht nur in der Forschung unterstützt, sondern gab mir auch die Freiheit und die Motivation dieses Ziel zu erreichen.

Genauso geht mein Dank an Professor Dr. Oliver Deussen, der immer begeistert von meiner kontroversen Forschung war und mich motivierte diese Richtung beizubehalten.

Mein ganz spezieller Dank geht an Dr. Andreas Stoffel, der mich in entscheidenden Momenten unterstützte und für Diskussionen und Rat immer tatkräftig bereit stand, als ich Hilfe brauchte.

Mein Dank gilt ebenfalls meinen Kollegen am Lehrstuhl Datenanalyse und Visualisierung für die stets gute Zusammenarbeit und Arbeitsatmosphäre. Insbesondere gilt mein Dank meinen lieb gewonnen Kollegen Juri Buchmüller, Dominik Jäckle, Halldór Janetzko und Florian Stoffel für ihre Unterstützung und ihre Ratschläge.

Letztlich gilt mein Dank meiner Familie, die mich immer moralisch unterstützt hat und in besonderem Maße meiner Freundin Svenja Simon für ihre private wie fachliche Unterstützung.

Nicht zu vergessen ist auch die Deutsche Forschungsgemeinschaft, die meine Dissertation im Rahmen des folgenden Projekts finanziert hat: Das Graduiertenkolleg GK-1042 "Explorative Analyse und Visualisierung großer Datenräume".

Contents

Für meine Eltern.

1 Introduction

"How can we encode information with color *effectively*?" is the central question of this thesis. The answers to this question form the main contributions of this thesis:

> a novel *definition* of effectiveness and novel *quality metrics* for color encodings,
>
> novel *guidelines and methods* to effectively encode information with color,
>
> and novel *methodologies and methods* which ensure that the *individual* user *perceives* the information of encolored data effectively *without any bias such as contrast effects*.

Thus, the word *effective* plays a major role in this thesis. However, to understand "why" we are using color to encode information and what "effectiveness" means in this context, we first need to understand "why" there is a need for visual data analysis.

> The methods of this thesis are optimized for LCD displays. If there are any doubts about images, please refer to the electronic version of the thesis. All images are of high resolution and are zoomable.

1.1 Visual Data Analysis

Data analysis in general has the aim to generate knowledge out of data. Several closely related fields such as statistics and data mining provide methods for knowledge generation. In order to show the value of visual data analysis, we first need to discuss what these fields can provide and how the human can interact with the methods to gain knowledge from data.

Descriptive statistics aim to model a known behavior or pattern that was observed — or is within the data. Inferential statistics aim to validate or reject hypothesis and models of a larger populations based on observed data given as representative sub-populations (sample data). For example, we could gather data about cars and record their "miles per gallon" (MGP), "horsepower" (HP), weight, origin, and number of cylinders (data set of Quinlan (1993)). If we want to analyze the relation of MPG and HP of cars, we could state the hypothesis that "there is a negative correlation between HP and MPG" (in other words: if a car has more horsepower, it will need more gasoline than a car with less horsepower). Therefore, we could test if the hypothesis holds for our sample of cars with, e.g., linear regression, which would confirm the hypothesis with statistical significance ($p < 0.0001$) and produce a model, e.g., $MPG \approx 40 - 0.16 \cdot HP(\pm 0.72)$. We could do the same analysis for the number of cylinders resulting in a similar model ($MPG \approx 43.61 - 3.66 \cdot cylinders(\pm 0.82)$). These models allow us to infer attributes, even for cars that were not included in the data set. For instance, we can calculate how many miles a car can drive per gallon based on HP or the number of cylinders.

The question is, however, "do we have *further* questions"? Is there anything unexpected in the data set, for which we cannot verbalize a question or define a concrete task to find the answer? To answer such questions, we need to *explore* the data. Information visualization is known to support such exploratory approaches. One of the most successful and accurate visualizations is

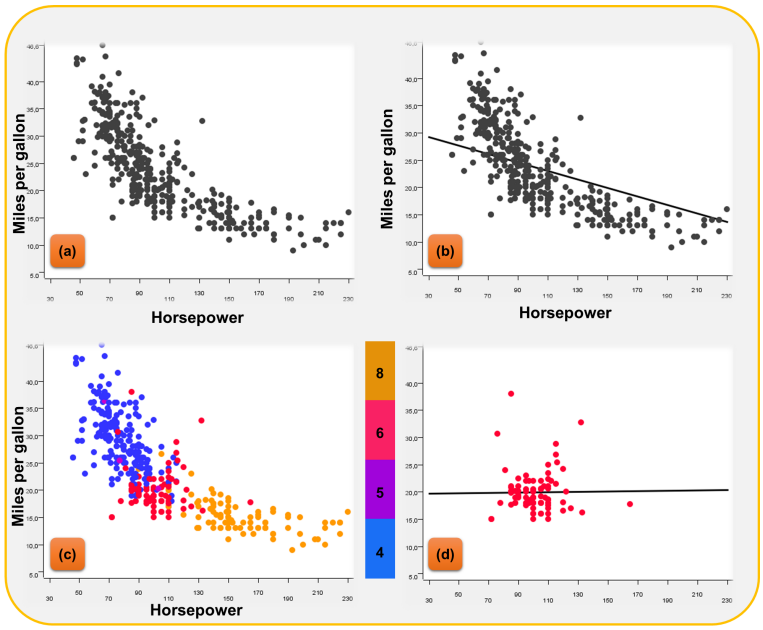


Figure 1.1: The images were created and statistics were calculated with KNIME (Berthold et al., 2007). *Visual data analysis of the car data set of Quinlan (1993). (a) The car data set is visualized in scatter plots with miles-per-gallon (MPG) on the y-axis and horsepower (HP) on the x-axis. (b) MPG decreases linearly with increasing HP (negative correlation). (c) Color is added to encode the number of cylinders. The red points form a cluster that seems to correlate positively, which is unexpected due to the global negative correlation. This implies that cars with six cylinders show a different behavior than other cars in this data set . (d) This is confirmed by applying statistics and visualizing the correlation of MPG and HP only for cars with six cylinders.*

the scatterplot. In Figure 1.1 (a), the car data is visualized with HP on the x-axis and MPG on the y-axis. We directly perceive the negative correlation (Figure 1.1 (b)) since our visual system aims to pre-attentively group visual objects according to similar visual features (e.g., dots in close proximity). Further, we can use color to encode the number of cylinders for each car (Figure 1.1 (c)). In this visualization, we perceive a similar pattern. Low numbers (blue, violet) on the top-left and high numbers (red, orange) on the bottom right. However, our eye pre-attentively groups the cars not only by their spatial position but also by color. A closer look reveals that the red dots do not follow the same negative trend as the other encolored dots. These are the cars with 6 cylinders, which

seem to be different. So, "is there a difference between cars with 6 cylinders and the other cars?". We can only guess from the visualization but cannot provide facts. However, this can be answered with statistics, e.g., calculating linear regression on cars with six cylinders only reveals that there is in fact no correlation ($p > 0.9$) between HP and MPG (Figure 1.1 (d)) for these cars, which is an unexpected finding.

Note, that the sample size of the data set is not representative and this finding does not infer causality. This example just aims to show how to raise questions by identifying unexpected findings with visual analysis.

Statistics share a founded mathematical base and if the analyst knows what to ask, these methods will provide an answer. The same is valid for automatic methods of exploratory data analysis (data mining). Clustering algorithms will detect expected and unexpected clusters, but they will always detect some specific kind of cluster. Association rule mining will detect expected and unexpected associations between items (dimensions) but always determine associations. Further, sometimes statistics can be extremely wrong if the model does not capture all important features of the observation. For example, Anscombe (1973) produced several obviously different data sets illustrated in Figure 1.2. We clearly perceive the difference visually. The results of the statistics, however, conclude that these different data sets are equal. This shows also a danger of statistics and automatic methods. Thus, we should not "blindly" trust statistics.

Statistics and data mining methods alone work extremely well if the analyst knows precisely what she/he is seeking for and can formulate questions *precisely* (Fekete et al., 2008). However, to find the right questions to asked is key for exploratory data analysis, where the analyst is faced with ill-defined problems. Automatic methods fail in some scenarios because they lack in flexibility, creativity, and general world knowledge to find answers for unsaid questions and unexpected findings. Visual analytics aims to integrate the user in the analysis loop to overcome these issues. It, thereby, facilitates our fast perception and recognition of unexpected patterns within visualized data and integrates, thereby, the knowledge and creativity of the human in the analysis process.

Anscombe's Quartet: Raw Data

	1		2		3		4	
	X	Y	X	Y	X	Y	X	Y
	10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
	8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
	13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
	9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
	11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
	14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
	6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
	4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
	12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
	7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
	5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89
Mean	9.0	7.5	9.0	7.5	9.0	7.5	9.0	7.5
Variance	10.0	3.75	10.0	3.75	10.0	3.75	10.0	3.75
Correlation	0.816		0.816		0.816		0.816	

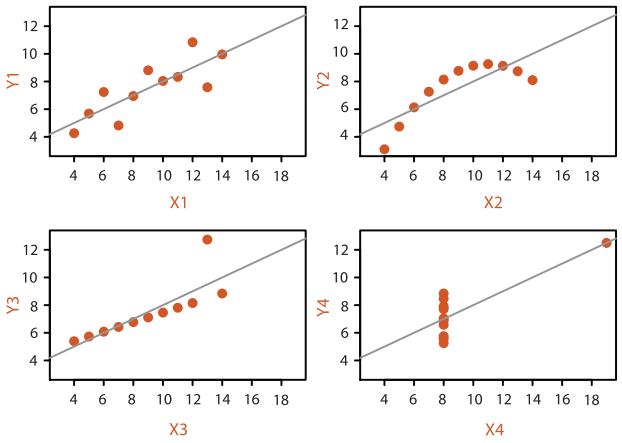


Figure 1.2: Data set of Anscombe (1973). Image taken from Munzner (2014). *Four obviously different data sets, for which statistics estimate the same mean, variance, and correlation.*

But how can we build data visualizations that facilitate the pre-attentive detection and recognition of patterns? — To answer this question and to assess human perception for data analysis, we need to understand which *visual variables* can be used to encode data.

1.2 Visual Variables and the Special Role of Color

If the visualization designer manages to visualize abstract data such that the user can simply perceive the relevant information within the data set, cognitive processes can be omitted, which makes visual data analysis more effective and efficient. The central question is "how" to achieve this.

One of the main challenges of visualization research is to find means to encode data with visual primitives that are pre-attentively processed. Ware (2012) describes that our visual cognition system processes the incoming visual information in three stages. Low-level features such as position, color, and orientation are extracted from the visual scene to form "feature maps" in the first level. These feature maps are then analyzed in the second stage that aims to detect visual patterns such as the area and texture of regions, as well as patterns of orientation and motion. These visual patterns are then connected to recognize objects in the third stage, which is driven by cognitive processes (steered by attention) and is, therefore, very slow. However, the first and second stages are rapidly and (predominantly) unconsciously processed such that we can efficiently capture visual information. However, we cannot control these processes even if we aim to focus on different aspects in the scene. These low-level features and visual patterns, therefore, form the elementary building blocks for encoding data visually and for pre-attentive perception of information (the so called "visual variables"). Some visual variables carry magnitude information and can encode ordinal and quantitative data, such as the length or size of patches. Others carry categorical information such as shape and can, therefore, be used for encoding nominal data.

Color is one of the most important and effective visual variables (Figure 1.3). It can be combined with any other visual variable to enrich data visualizations

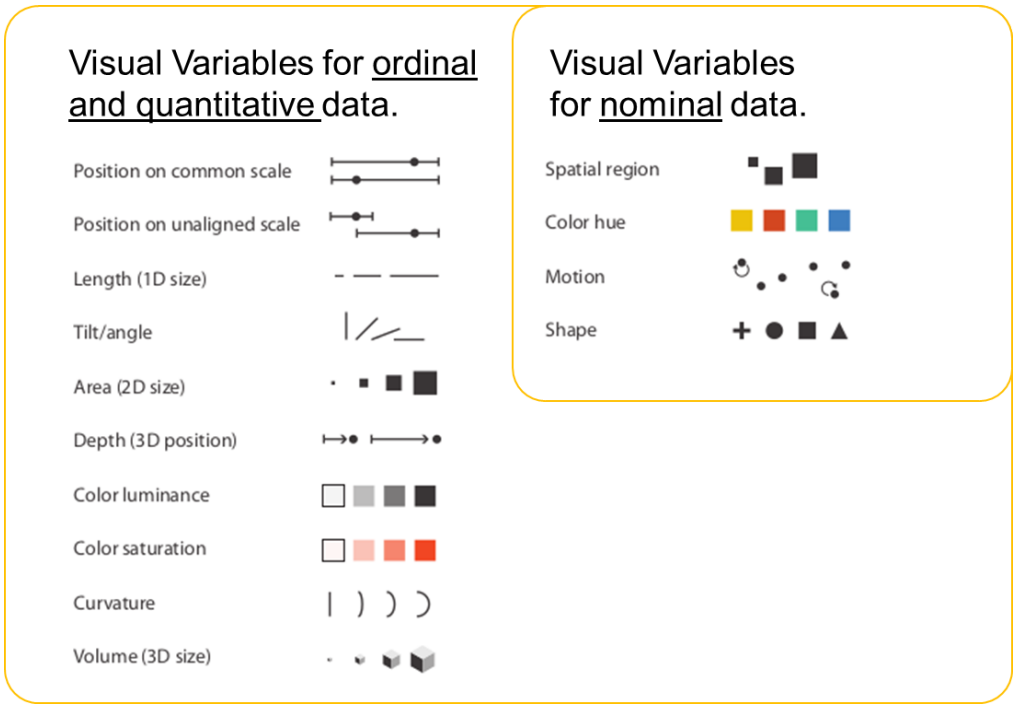


Figure 1.3: Image adapted from Munzner (2014). *Visual variables that allow pre-attentive perception of nominal, ordinal and quantitative information.*

without using additional visualization space; thus, it enables high scalable visualizations. The ability of the visual cognition system to discriminate huge amounts of colors in the earliest stages of visual perception is the base of other visual patterns. We amplify color contrasts to detect edges and group encolored patches to recognize texture, shapes, and areas of objects. Thus, discriminating and grouping objects by position and color is more efficient than with other visual variables. Color provides orthogonal channels to encode categorical information with hue (red, green, blue, etc.) and magnitude information with saturation and luminance (Figure 1.3). Our vision system provides a natural perceptual kernel such that we can relate colors, e.g., violet is more similar to blue than to green. With these orthogonal channels, color is among the few visual variables that can be designed by combining the channels to support more complex attribute types and analysis tasks.

However, color can also encode more than just nominal, ordinal, and quan-

titative data. It is a unique feature of color that some colors carry sentiment or emotion (e.g., green is positive, red is negative) (Kuhbandner and Pekrun, 2013) and semantics (water is blue, red is danger) (Lin et al., 2013). Language is very strong in the evolution and development of color perception. Research showed that language and verbal interference (Roberson and Davidoff, 2000) has significant impact on how we perceive and recognize color. Therefore, we can easily name objects by their color and communicate findings to other people. However, cultural and domain differences make the design of colormaps a challenging task.

Because of its complexity, color perception was and is one of the core research topics in information visualization and there are still many open questions on "how to *effectively* use color to encode information".

1.3 Open Questions and Contributions of this Thesis

Obviously, the usage of color  is more than just assigning few colors to single data values and there are many ways to do it wrong. Often "standard" colormaps are used without any justification. For example, the "rainbow colormap" (above) is widely used and has almost become a standard in visualization toolkits, although it is known to be misleading.

There are many guidelines for color mapping, however, some propose conflicting rules-of-thumb and important results of perceptual user studies are not considered in the state-of-the-art colormap design (Figure 1.4). The question is "why?". For example, there is one group of researchers who aim to generate colormaps that are as "expressive" as possible, which means that they provide *many* distinct colors to encode *many* data values. This is typically achieved by

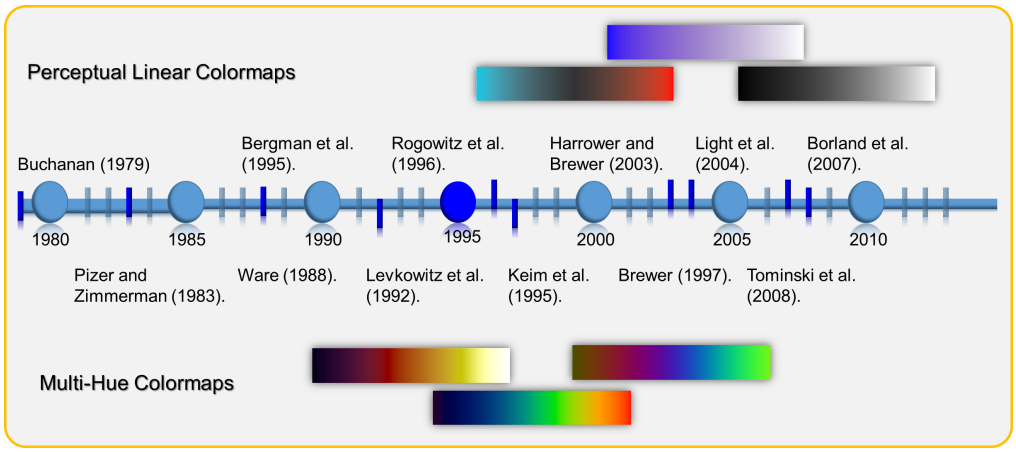


Figure 1.4: *Excerpt of colormap research. Two groups propose conflicting guidelines and goals for designing colormaps.*

varying over hues with linear increasing intensity (from dark to bright colors, see bottom of Figure 1.4). These researchers provide evidence that these multihue colormaps perform very well in their particular applications. However, there is another group claiming that these colormaps are misleading because they are not perceptual linear and do not encode data in-line with human perception. They are precisely describing harmful properties of multihue and rainbow colormaps for their analysis tasks. Theoretically, it is clear that a colormap that varies over multiple hues is bad for encoding *quantitative data* (which requires representing *ordered* values) because we simply *cannot order hues*. However, Ware (1988) performed an experiment and found that the rainbow colormap outperforms other (perceptual linear and perceptually ordered) colormaps in *reading metric quantities*. *"Why is the rainbow colormap effective for reading quantitative data if its perceptual properties should be misleading for this task?"*

The reason is that the participants did not *compare* quantitative values but did *identify* quantities. We claim: the *effectiveness* of a colormap *depends on the analysis task* that is performed with the colormap. This implies that both of the mentioned groups are correct. The controversy results from applying colormaps in analysis tasks that are not appropriately supported by the colormaps. Therefore, we *define the effectiveness* of color encodings depending on analysis

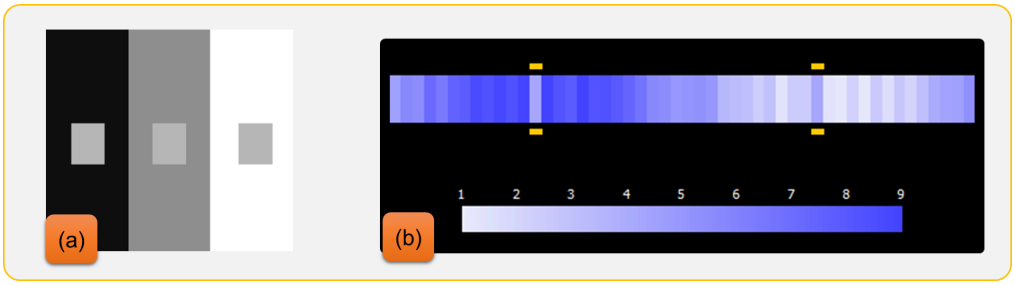


Figure 1.5: *Contrast effects can bias our perception of color. (a) The gray patches share the same gray value but are perceived differently. (b) The encoded data values appear differently to a human analyst, however, they encode the same data value (4).*

tasks in order to bridge the controversy in the research field and to *provide novel methods and guidelines* for effective color encodings based on this definition.

However, one of the most critical biases and unsolved problems in information visualization are contrast effects (Figure 1.5). These effects cannot be overcome by just designing effective color encodings because color appearance is relative. How a color encoded data value is perceived depends on the surround. For example, a gray patch is perceived brighter on a black background than on a white background (Figure 1.5 (a)). If we would compare the colors of Figure 1.5 (b), we would assume that the values are different, however, they are identical. Ware (1988) found that contrast effects can bias users reading color encoded data by up to 20%. Even if we design effective color encodings that fit optimal to the analysis task and apply the existing guidelines and rules-of-thumbs for avoiding contrast effects, the analyst may still be biased in the final visualization, because *how the color of a data object is perceived depends on its surround in the rendered image*. We cannot compensate for these effects in advance, even if we are (cognitively) aware of this issue. Because these effects are caused in the earliest stages of visual perception, of which we have no control. But how can we compensate for these effects and accurately visualize data with color?

To capture *how the individual user perceives* a data visualization and how the above question can be addressed by *automatically adapting* the visualization to provide faithful color encodings is the core contribution of this thesis.

In-line with the conventions of scientific writing, this thesis uses the words “I” and “we” as synonyms and does not distinguish between contributions that were developed 100% by myself or contributions that were collaboratively developed. Please refer to Section 1.5 (p. 18) for the specification of contributions and work distribution of co-authors of the publications used in this thesis.

1.3.1 Definition of Effectiveness for Color Encoding

Effectiveness is an ambiguous word and has to be defined in the context in which it is applied. We gathered and interpreted the guidelines of the state-of-the-art for color mapping in information visualization to formulate a verbal definition, which may also be valid for other visual variables and visualizations but is focused predominantly on color. This definition structures this thesis and clarifies the provided contributions.

Note, that this definition is extended and formally described by precise quality metrics after analyzing the perceptual foundations of color vision and defining requirements for analysis tasks in Chapter 2 (p. 31). In our verbal definition, effective color encodings are ...

pre-attentive. Color encodings must be in-line with the perception of a user such that color encoded values can be perceived pre-attentively without cognitive processes (Munzner, 2014). For instance, categorical information is encoded with categorical colors and continuous information is encoded with perceptual continuous gradients of color, which let us *perceive* magnitudes and differences in magnitudes.

semantic consistent. Since color also carries semantics and sentiment, cultural as well as domain differences have significant impact on the perception of color. Therefore, the color encoding must also be in-line with the mental model of domain users and needs to consider also the culture and preferences of target users (Lin et al., 2013).

faithful. Colors must represent the data accurately (Rogowitz et al., 1996), which means that the color encoding must reflect the properties of the data. For instance, categorical data must be encoded with perceptually distinct colors such that the user does not mix up different categories. Similar, the color encoding of quantitative data must ensure that distances in data space are equal to perceived distances. Contrast effects do have a critical influence on this property, since they bias analysts in accurately reading and comparing the encoded data value.

expressive. The color encoding must visualize *all* the information within the data (but only the information, see *faithful*) (Mackinlay, 1986). For instance, for 12 different categories it must provide 12 perceptually distinct colors and for quantitative data it must provide as many colors as possible within the data range.

supporting the analysis task. Visualization systems must support the analysis tasks (Brehmer and Munzner, 2013). The requirements for elementary and synoptic tasks such as *localization*, *identification* and *comparison* are quite different (Tominski et al., 2008). For instance, in *localization* the analyst aims to find certain values on the display and the color encoding should support the search; in *comparison* the analysts focuses on two or more objects on the screen to perceive their absolute or relative differences. Real analysis tasks also require the combination of these elementary tasks, e.g., the analyst aims to *localize* and *compare* data values. Therefore, the color encoding must satisfy the requirements of the target analysis tasks.

1.3.2 Contributions & Structure of the Thesis

Requirements, Guidelines, and Methods for Designing *Effective* Color Encodings

are presented in Chapter 2 (p. 31). In Section 2.1 (p. 34), the challenges for color encodings are further motivated and the discussion of related work in Section 2.2 (p. 37) reveals that there is a difference in encoding single data dimensions and (high dimensional) data relations, which divides the chapter into two parts.

The first part in Section 2.3 (p. 43) analyzes the perceptual foundations of color vision to *effectively* encode single data dimensions with color. Furthermore, since the state-of-the-art fails to provide guidelines for supporting combined analysis tasks with color, novel requirements for (combined) elementary analysis tasks are defined, and novel precise quality metrics are provided to measure the effectiveness of colormaps. Further, Section 2.3 (p. 43) introduces novel guidelines to design effective colormaps for single dimensions, as well as the tool *ColorCAT*, which guides visualization experts in the design of color encodings and thereby enables the expert to match the color encoding with the mental model, preferences, and culture of the target user and application. The usefulness of *ColorCAT* is illustrated with use cases of adverse drug event detection and monitoring applications for security visualizations.

Section 2.4 (p. 76) builds the second part of this chapter. It defines novel requirements for (combined) synoptic analysis tasks and provides novel precise quality metrics to measure the effectiveness for encoding data relations with color. Based on these quality metrics, Section 2.4 (p. 76) introduces a novel optimization algorithm that projects high dimensional data into perceptual uniform color spaces, which maximizes the *effectiveness* of encoding data relations. A use case of visually analyzing large volumes of smart grid measurements illustrates the usefulness of this method. Further, a quantitative user study reveals that the method outperforms the state-of-the-art.

This chapter, thereby, satisfies the requirements of *pre-attentiveness*, *semantic consistency*, and *the support of analysis tasks* for providing effective color encodings. However, *faithfulness* and *expressiveness* can only partially be satisfied since they depend on how the user perceives the rendered visualization. The issues of contrast effects (biasing faithfulness), contrast sensitivity (limiting expressiveness), and just-noticeable-differences (limiting expressiveness) can only be overcome in the final rendered visualization with the contributions described in the following.

A Methodology and Method to Compensate for Contrast Effects Preserving the *Faithfulness* of Color Encodings

are provided in Chapter 3 (p. 105). The impact of contrast effects on visual data analysis and the need to compensate for them is illustrated in Section 3.1 (p. 109). Further, Section 3.2 (p. 111) reviews related work and discusses why the state-of-the-art guidelines to avoid contrast effects fail to provide *faithful* color encodings.

Section 3.3 (p. 112) presents a novel methodology to compensate for physiological biases based on perception models and optimization algorithms. Further, a novel method to compensate for contrast effects is provided, which is evaluated in Section 3.4 (p. 126). Two experiments with 40 participants revealed that this method doubles the accuracy of participants reading and comparing color encoded data values. In Section 3.5 (p. 135) the method is applied to visualizations of other authors and it is shown that contrast effect compensation increases the *faithfulness* of color encodings.

The method and methodology presented in this chapter is the base for Chapter 4 and Chapter 5. In combination with contrast effect compensation, the guidelines and methods of Chapter 2 can provide *effective* color encodings for the average human observer and common environment conditions. However, as a general approach, the method of contrast effect compensation does not account for individual differences in contrast perception (see next paragraph). Further, it does only preserve the information that is within the rendered visualization. Therefore, it is not able to compensate for effects that are due to global

color mappings, which may mask important local information and data patterns that are not recognizable in the image (see Chapter 5).

An Efficient Algorithm for Contrast Effect Compensation and Methods for Personalizing Contrast Perception

are provided in Chapter 4 (p. 141). Section 4.1 (p. 144) raises the need for efficient algorithms to compensate contrast effects for interactive visualizations and interactive experiments that allow, for example, the personalization of contrast effects.

A novel efficient algorithm based on surrogate models, efficient optimization, and massive parallelism is presented in Section 4.3 (p. 149). This algorithm is applied within the methodology of Section 3.3 (p. 112) and reduces the runtime from 4 minutes to 360ms compared to the method of Chapter 3 (p. 105). This efficient algorithm is applied for personalizing contrast effect compensation.

Section 4.4 (p. 158) provides a novel perception model and methods to capture the individual differences of color and contrast perception. This allows for adapting contrast effect compensation to an individual target user. Further, novel methods are introduced to adapt contrast effect compensation to different environment settings, e.g., to adapt to viewing distance, ambient light, and display devices.

In Section 4.5 (p. 170), the method of personalization is evaluated in a user study that measured the impact of contrast effects and their compensation with different perception models. This study shows that without contrast compensation, user make errors up to 24% caused by contrast effects, which is significantly reduced to 14% with the standard perception model, but is further significantly decreased to 10% with personalized models.

Methods to Boost "Hidden" Information to Enhance the *Expressiveness* of Color Encodings

are presented in Chapter 5 (p. 180). Effective color mappings may be effective globally (for the whole visualization representing overviews of data) but may not be effective locally. For example, global color mappings for data sets with

high variations often map local variations to a small value range, which cannot be perceived by the human analyst because the human eye is not sensitive enough. Thus, there may be important local patterns in the data that may become invisible due to global color mappings, which limits the *expressiveness* of color encodings.

Section 5.1 (p. 183) motivates this problem and introduces the idea of *local adaptive color mapping*. Section 5.2 (p. 186) shows that existing techniques fail to provide *expressive* and *faithful* mappings. Section 5.3 (p. 189) provides two novel methods and heuristics for this problem: supervised *perceptually optimized color boosting* and unsupervised *local edge preserving color mapping*. Both methods are evaluated against each other and the state-of-the-art. The results reveal that both methods outperform the state-of-the-art since they are able to reveal more hidden *local data structures* but preserve the faithfulness of the color encoding at the same time. For example, in data visualizations where an effective global color scheme hides 75% of the local data patterns, the supervised boosting method reveals at least 96% of the local data patterns with a color distortion of only 1.28%. The state-of-the-art only reveals 47% with a distortion of 25%. Section 5.4 (p. 213) shows the applicability of these methods in different use cases.

1.4 Citation Conventions

Major parts of this thesis appeared in journal and conference publications that I authored or co-authored (see Section 1.5). To distinguish these publications from references, a different reference style is used.

My publications are numbered with arabic numbers, for instance, [14]. References are cited with author names and year, for instance, (Sedlmair et al., 2014) or in-line: . . . the method presented by Lee et al. (2012) . . .

As it is the accepted scientific practice and guidelines of the research commu-

nity in computer science, all the major contributions of this thesis are published in journals and conference proceedings. I retain the copyright of all my publications that are used in this thesis. Parts of the chapters that appear verbatim in my publications are written by myself. Section 1.5 lists the publications that I authored or co-authored and clearly specifies the contribution and work distribution among the co-authors. At the beginning of each chapter I state the publications it is based on. For instance:

> This chapter is based on the following publications and major parts of the sections also appeared in the following publications:
>
> [11] Sebastian Mittelstädt and Daniel A. Keim. *Efficient Contrast Effect Compensation with Personalized Perception Models*. Computer Graphics Forum, 34(3):211–220, 2015.
>
> [14] Sebastian Mittelstädt, Andreas Stoffel, and Daniel A. Keim. *Methods for Compensating Contrast Effects in Information Visualization*. Computer Graphics Forum, 33(3):231–240, 2014.
>
> ---
>
> For the division of responsibilities and work, as well as a statement of contributions in these publications, please refer to Section 1.5 (p. 18).
>
> The following contributions go beyond the published work:
>
> 1. Integration of contrast sensitivity into the perception model, which models the view distance of observers.
> 2. Methodology for display dependent perception models for adapting contrast effect compensation to different output devices.
> 3. Methods to boost high-frequency information to account for contrast sensitivity.
> 4. Methods to compensate contrast effects depending on the view distance.

Other parts are based on my publications, but the text is paraphrased and extended. Contributions beyond the published work are listed in the beginning of each chapter. Paragraphs that are based on the contributions (and text) of co-authors are quoted, italicized, and cited accordingly.

1.5 My Publications & Contribution Specification

The following list specifies the contribution and work distribution among the authors of the papers that I authored or co-authored and are used in this thesis. The papers are ordered by the importance of the contributions in this thesis.

[14] Sebastian Mittelstädt, Andreas Stoffel, and Daniel A. Keim. *Methods for Compensating Contrast Effects in Information Visualization*. Computer Graphics Forum, 33(3):231–240, 2014.

The main research question "How to compensate for contrast effects?" was identified in a discussion with Keim. I defined and developed the contributions: 1) A method for compensating physiological color effects based on color appearance models and optimization algorithms that can be used on any data visualization as a post-processing step; 2) A definition of the optimization goal and the corresponding perceptual metrics; 3) A general heuristic to approximate the gradient of compensation; 4) An evaluation of the perception model and the compensation, based on realistic tasks and data. All sections were written by myself. The implementation was performed by myself and I designed and performed the user study. Stoffel reviewed the paper drafts. Keim supervised the paper project and commented on paper drafts.

[11] Sebastian Mittelstädt and Daniel A. Keim. *Efficient Contrast Effect Compensation with Personalized Perception Models*. Computer Graphics Forum, 34(3):211–220, 2015.

The main research questions "How to personalize contrast effect compensation?" and the research challenge to provide "an efficient algorithm for interactive visualizations" was identified by myself. I defined and

developed the contributions: 1) An efficient method to compensate for contrast effects; 2) methods to personalize contrast effect compensation; 3) an evaluation of personalized contrast effect compensation with a user study. All sections were written by myself. The implementation was performed by myself and I designed and performed the user study. Keim supervised the paper project and commented on paper drafts.

[10] Sebastian Mittelstädt, Dominik Jäckle, Florian Stoffel, and Daniel A. Keim. *ColorCAT: Guided Design of Colormaps for Combined Analysis Tasks*. In Proceedings of the Eurographics Conference on Visualization, pages 115–119. The Eurographics Association, 2015.

The main research question "How to provide color encodings for combined analysis tasks?" was identified by myself. I defined and developed the contributions: 1) A definition of requirements for different analysis tasks and their combinations; 2) Quality metrics for one dimensional colormaps to support these requirements and; 3) Color-blind safe color maps for each task combination; 4) The tool *ColorCAT* for guided design of colormaps. All sections were written by myself. The implementation was performed by myself. The authors Jäckle and Stoffel reviewed the paper drafts. Keim supervised the paper project and commented on paper drafts.

[8] Sebastian Mittelstädt, Jürgen Bernard, Tobias Schreck, Martin Steiger, Jörn Kohlhammer, and Daniel A. Keim. *Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis*. In Proceedings of the Eurographics Conference on Visualization (EuroVis 2014), pages 91–95, 2014.

The main research question "How to encode high dimensional data with color?" was identified in a discussion of the authors Mittelstädt, Bernard, Schreck, and Steiger. I defined and developed the contributions: 1) A generalization and extension of the method of Kaski et al. (2000) to map high-dimensional data to perceptual uniform color spaces; 2) Efficient heuristics for practical use; 3) Perceptual color mapping quality metrics

and their combinations for visual analysis tasks; 4) An evaluation of the method with a user study. The implementation was performed by myself and I designed and performed the user study. All sections were written by myself. The authors Bernard, Schreck, and Steiger reviewed the paper drafts. Kohlhammer and Keim supervised the paper project and commented on paper drafts.

[15] Sebastian Mittelstädt, Andreas Stoffel, Tobias Schreck, and Daniel A. Keim. *Analysis of Local Data Patterns by Local Adaptive Color Mapping*. Presented at the IEEE Conference on Visualization (poster paper), 2014.

The main research question "How to locally adapt color mappings to enhance the visibility of local data patterns" was identified in a discussion with Keim. I defined and developed the contribution: A color boosting algorithm that locally adapts the color mapping for important data structures and guarantees the visibility of important data points. All sections were written by myself. The implementation was performed by myself. The other authors reviewed the paper drafts. Keim supervised the paper project and commented on paper drafts.

[1] Jürgen Bernard, Martin Steiger, Sebastian Mittelstädt, Simon Thum, Daniel A. Keim, and Jörn Kohlhammer. *A survey and task-based quality assessment of static 2D color maps*. In Proceedings of SPIE 9397, Visualization and Data Analysis, page 93970M, 2015.

The main research challenge "to survey and to assess the quality of static two-dimensional colormaps" was identified in a discussion of the authors Bernard, Steiger, and Mittelstädt. Bernard defined the contributions: 1) a survey of the most prominent static 2D colormaps; 2) a taxonomy of task-based requirements for the use of 2D colormaps for multivariate data; 3) novel quality measures based on the requirements of the established task model; 4) combinations of relevant quality criteria for analysis tasks. Contributions (1) and (3) were mainly developed by Bernard, Steiger, and Thum. I defined task-based requirements for the use of 2D colormaps for

contribution (2) and further mapped combinations of quality metrics to the according analysis tasks for contribution (4). The implementation was performed by Bernard, Steiger, and Thum. The requirement definitions of Section 3.1 “Requirements for Applying 2D Colormaps based on a Set of Analytical Tasks” was written by myself and I contributed the text and formula of the quality metrics of “Visual Importance” in Section 3.2 “Quality Assessment Measures for 2D Colormaps”. All authors reviewed the paper drafts. Kohlhammer and Keim supervised the paper project and commented on paper drafts.

[16] Sebastian Mittelstädt, Xiaoyu Wang, Todd Eaglin, Dennis Thom, Daniel Keim, William Tolone, and William Ribarsky. *An integrated in-situ approach to impacts from natural disasters on critical infrastructures*. In Proceedings of the 48th Hawaii International Conference on System Sciences, pages 1118–1127. IEEE, 2015.

The main research challenge to provide a “visual analytics system for different levels of crisis response for critical infrastructures” was identified by all the authors. I defined the contributions: To provide a visual analytics system that: 1) supports all levels of crisis response with specialized equipment and visualizations for control rooms and mobile devices; 2) combines multiple critical infrastructures and social media by information abstraction; 3) enables interactive simulation and visualization of the subsequent development of a crisis; 4) enables interdisciplinary and distributed teams to understand and react on crisis situations. I provided the concepts for (1)–(4). I developed the visual analytics system for monitoring several critical infrastructures in (2) and the integration and interactive simulation of the subsequent development of a crisis (3). All sections were written by myself. The implementation was performed by Alexander Jäger. The other authors reviewed the paper drafts. Keim and Ribarsky supervised the paper project and commented on paper drafts.

[12] Sebastian Mittelstädt, David Spretke, Dominik Sacha, Daniel A. Keim, Bernhard Heyder, and Joachim Kopp. Visual Analytics for Critical Infras-

tructures. In Proceedings of the International ETG-Congress 2013; Symposium 1: Security in Critical Infrastructures Today, pages 1–8. VDE, 2013.

The main research challenge to provide "visual analytics for the analysis of multiple interdependent critical infrastructures" was identified by all the authors. I defined the contributions: To provide a visual analytics system that: 1) combines multiple critical infrastructures by information abstraction; 2) perceptually highlights important events; 3) enables an interdisciplinary team to understand crisis situations; 4) and reveals domain details and controls on demand All sections were written or significantly revised by myself. The implementation was performed by Alexander Jäger. The other authors reviewed the paper drafts. Keim supervised the paper project and commented on paper drafts.

[9] Sebastian Mittelstädt, Ming C. Hao, Umeshwar Dayal, Meichun Hsu, Joseph Terdiman, and Daniel A. Keim. *Advanced Visual Analytics Interfaces for Adverse Drug Event Detection*. In Proceedings of the Working Conference on Advanced Visual Interfaces, pages 237–244, 2014.

The main research challenge to provide a "visual analytics system for adverse drug event detection" was identified by all the authors. I defined and developed the contributions: 1) A visual analytics approach to access massive volumes of events by interactive relevance filtering; 2) Detection and validation of low frequency events by enhanced statistical computations and interactive analysis; 3) Elimination of confounding effects by using discriminative heuristics. All sections were written by myself. The implementation was performed by myself. The other authors reviewed the paper drafts. Keim supervised the paper project and commented on paper drafts.

[21] Martin Steiger, Jürgen Bernard, Sebastian Mittelstädt, Hendrik Lücke-Tieke, Daniel A. Keim, Thorsten May, and Jörn Kohlhammer. Visual *Analysis of Time-Series Similarities for Anomaly Detection in Sensor Networks*. Computer Graphics Forum, 33(3):401–410, 2014.

The main research challenge "to design a visualization system for interactive pattern analysis in univariate sensor networks" was identified by the other authors. I contributed a two-dimensional colormap and the according description in Section 4.1 on creating such a colormap. The implementation was performed by the other authors. All authors reviewed the paper drafts.

[7] Halldór Janetzko, Florian Stoffel, Sebastian Mittelstädt, and Daniel A. Keim. *Anomaly Detection for Visual Analytics of Power Consumption Data*. Computer & Graphics, 38:27–37, 2014.

The main research challenge "to design anomalies in power consumption data" was identified by the other authors. I contributed the color encoding with highlighting and the according text of Section 4.1. The implementation was performed by the other authors. All authors reviewed the paper drafts.

[18] Lin Shao, Sebastian Mittelstädt, Ran Goldblatt, Itzhak Omer, Peter Bak, and Tobias Schreck. *StreetExplorer: Search-based exploration of urban street networks.* Submitted to the International Conference on Information Visualization Theory and Applications, 2016.

The main research challenge "to apply visual analytics for search-based exploration of street network" was identified by the other authors. The contributions were defined by all authors: 1) Similarity functions to rank and compare street network properties. 2) Helpful interaction functions, which allow the user to interactively select local areas of interest. 3) A suitable color-mapping and boosting scheme, which allows visualizing local similarity to a user query in context of the overall feature distribution. 4) The application and evaluation with domain experts on real data and real existing street networks. I developed and implemented (3). Further, I authored the sections about color mapping and visual boosting. The rest of the paper was written by the other authors. All authors reviewed and commented on paper drafts.

1.6 Color Foundations & Analysis Tasks

Since this thesis discusses detailed issues and methods of color perception and encodings, this section aims to provide basics of color perception and color spaces. Further, it introduces a terminology that is used to describe analysis tasks by merging the task typologies and terminologies of Andrienko and Andrienko (2006) and Brehmer and Munzner (2013).

1.6.1 Color Perception & Color Spaces

Light consists of different wavelengths of energy that is absorbed by the eye. The perception of color is only the interpretation of the incoming light information by determining the different wavelengths of light and their magnitudes. The human eye has only receptors for red (long), green (middle), and blue (short) wavelengths of light. Every color can, therefore, be produced by emitting a combination of red, green, and blue light with varying intensity as described by the trichromacy theory (Fairchild, 2013).

Additive displays exploit the perception of the human eye and combine emitters (e.g., LEDs) to emit red, green, and blue light (display primaries) in every pixel to produce a broad subset of colors that the human eye can perceive (Figure 1.6). Therefore, additive color spaces such as RGB are cubical color spaces with three dimensions. Colors can be defined by setting the intensity of each emitter to a value within the physical range, which is typically normalized to [0,255]. The vector [255,255,255] in this space encodes white and [0,0,0] black.

Printed media cannot emit light but only reflect it. A white page reflects the whole incoming light. Green patches on a white page reflect green wavelengths

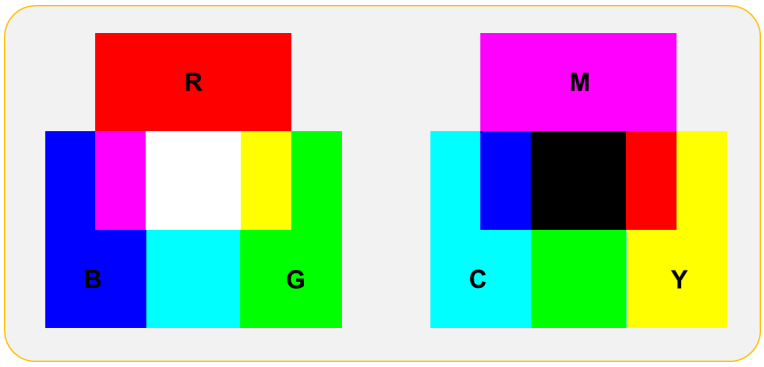


Figure 1.6: *RGB and CMYK color space.*

of light but absorb any other wavelength than green. Therefore, printed media uses subtractive color spaces such as CMYK (cyan, magenta, yellow, and black). Each of these subtractive dimensions absorbs certain wavelengths of light (Figure 1.6).

It is hard for the human to design colors with combining different primary colors, because it is not in-line with our perception. We perceive the hue of a color rather as a category (red, magenta, orange etc.) than as a combination of different primary colors (see Section 2.3.2 (p. 45)). Color spaces such as HSV, HSL, and HSI (Figure 1.7) aim to provide dimensions for designing colors according to our visual channels. For example, in HSV these dimensions are defined as hue (the color category, e.g., blue, cyan, lime), saturation (the vividness of colors), and value (the intensity of the emitted color). In HSL, the last dimension is defined by lightness, which "more" accurately describes how the lightness of the emitted color is perceived by the human. The advantage of the HSI color space (Keim, 2000) is that intensity and saturation are "more" orthogonal to each other (changes in intensity does not change the saturation) than in HSL, which allows creating colormaps that vary over hues with perceptual linear increasing intensity while preserving saturation (see Section 2.3.3 (p. 51) and Section 2.3.4 (p. 65) for more details).

All the above color spaces share the problem that color differences estimated in the color space do not correspond to perceptual differences. Therefore, researchers developed perceptual uniform color spaces such as CIELAB (CIE, 1978)

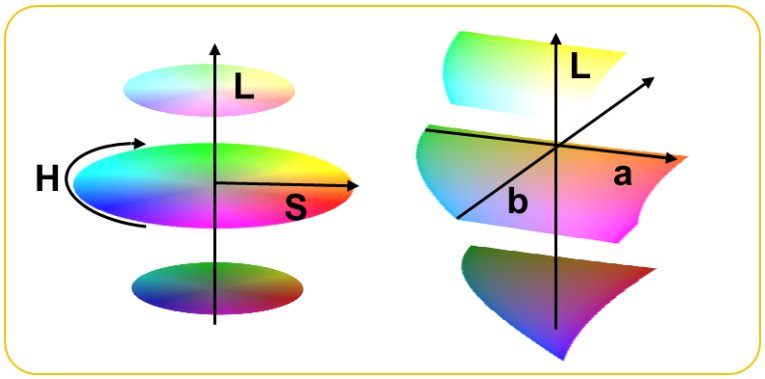


Figure 1.7: *HSL and CIELAB color space.*

and DIN99 (DIN, 2000). These spaces allow vector arithmetics that are in-line with human perception and color differences calculated by euclidean distance match perceived color differences (Eq. 1.1). According to the opponent process theory (Fairchild, 2013), these spaces typically represent colors with the dimensions lightness (L) and the two chromatic dimensions red-green (a) and yellow-blue (b) (Figure 1.7). DIN99 is an extension to CIELAB and redefines the color space to provide an accurate model of perceptual differences. Both color spaces are defined for small color differences ($\Delta E < 5$), and are inaccurate for big color differences.

$$\Delta E := \sqrt{\Delta L^2 + \Delta a^2 + \Delta b^2} \tag{1.1}$$

The CIECAM02 (Moroney et al., 2002) is the state-of-the-art color appearance model. It provides perceptual dimensions similar to the other color spaces and describes lightness, luminance, chroma, saturation, and hue (and many more perceptual dimensions). It also considers the ambient lightening conditions and perceptual processes such as chromatic adaptation (see Section 3.3.2 (p. 118) for more details). To overcome the issues of big color differences, Luo et al. (2006) provide perceptual uniform color spaces that are valid for small and big color differences.

More details on how humans perceive color and advanced color spaces are described in Section 2.3.2 (p. 45) and Section 3.3.2 (p. 118).

1.6.2 Analysis Task Typologies

Analysis tasks are important for our definition of effectiveness for color encodings. In the following, the task typologies of Brehmer and Munzner (2013) and Andrienko and Andrienko (2006) are revisited and merged in an intuitive terminology for this thesis.

Multi-Level Tasks.

Brehmer and Munzner (2013) classify different levels of analysis tasks. The highest level consists of the goals for data visualizations: to present information, to discover knowledge, to enjoy and produce information. The middle level describes four different types of *searching* for objects or their characteristics in the data visualization. The different types of *searching* are classified by the availability of location and characteristics. If the analyst knows the location of a target object, he/she can *browse* for its characteristics (e.g., the analyst reads the data values of a target object on the display). This highlights that the location of objects must be known *a priori*, before the characteristics of objects can queried on the lowest level of the task taxonomy. If the analyst searches for the occurrence of certain characteristics in the data, the analyst's task is to *localize* the objects of interest (e.g., to search for the objects that share the value 100). This highlights that the target characteristics must be known *a priori* to *localize* the target objects. In the *explore* task, neither the characteristics nor the location of target objects is known and the analyst *explores* the data and retrieves either location or characteristics of target objects. Once a single target is found, the user queries on the lowest task level for the characteristics or occurrences of certain characteristics on the display (*identify*). A set of targets can be *compared* or *summarized* by the user.

The task taxonomy of Brehmer and Munzner (2013) is easy to understand, applicable to many data analysis scenarios, and clearly defines the *explore* task, but there are also a few shortcomings. The query task of *comparing* two objects retrieves their similarity or difference. Thus, it retrieves the characteristics of

the comparison of two or more objects, for which the location on the display is known. However, if the location is unknown and the analyst queries for the occurrence of a certain "difference or similarity" between objects, he/she searches for the location of a "difference or similarity". We think that "relation-seeking" is a more intuitive description for this task, which is also used in the terminology of Andrienko and Andrienko (2006). Further, in the definition of Brehmer and Munzner (2013), the tasks "identification" and "comparison" can only be performed on single or few objects and only "summarize" can be performed on a set of data objects. However, the typology of Andrienko and Andrienko (2006) highlights that also these tasks can be performed on sets of objects, e.g., identify or compare clusters.

Elementary and Synoptic Tasks.

Andrienko and Andrienko (2006) distinguish elementary tasks that are performed on single data objects (e.g., read or identify the data value of one object) and synoptic tasks that focus on sets of objects (e.g., identify clusters in the data set). Single data objects as well as sets of objects can be looked-up, compared, or related in direct or indirect manner. A more intuitive terminology based on this typology is also provided by Tominski et al. (2008). Direct look-up can be considered as *identification*, which retrieves the characteristics of a data object. Indirect look-up can be considered as *localization*, which retries the location of objects with certain characteristics on the display. Direct *comparison* is the task of comparing the characteristics of two data objects. The difference to indirect comparison and *relation-seeking* is that the characteristics are not known *a priori* but the location. In *relation-seeking*, the characteristics are known but the analyst wants to search for the occurrence of the relation (e.g., search for the objects that have a difference of 50 in attribute X). These tasks can also be performed on sets of objects, since cluster can be *identified*, *compared*, and classes can be *localized* on the display. This complies to the *browse* (*identification*) and *locate* (*localization*) tasks of Brehmer and Munzner (2013), but is not clear that either the locations or the target characteristics must be known *a priori* to perform these tasks.

Merged Terminology

To merge the different terminologies (identify vs. identification vs. browse) and tasks for this thesis, we merge the typologies in the following. We distinguish between elementary (analysis of single objects) and synoptic tasks (analysis of sets of objects) following the typology of Andrienko and Andrienko (2006). While *identification* and *comparison* retrieve the characteristics of target objects (for which the location / visual reference is known), *localization* and *relation-seeking* retrieve the location on the display (visual reference) of objects that comprise the characteristics or relations the analyst is searching for. For all these tasks, the analyst requires knowing either the location of objects or their characteristics in order to perform these analysis tasks. For all these tasks we can define requirements for the design of visual variables since these tasks are well defined. However, Brehmer and Munzner (2013) highlight that the *explore* task is different since neither location nor characteristics of analysis targets may be known *a priori* and the analyst searches for both. Further, the analyst may be interested to identify and compare data objects in the exploration. Thus, this task is inherently ill-defined and it is hard to define specific requirements. However, we argue that exploration can be supplied by combining different elementary or synoptic analysis tasks and thus, the requirements for exploration can also be derived by the requirements of the elementary and synoptic tasks. We briefly summarize our task terminology and discuss requirements for elementary and synoptic tasks in Section 2.3.1 (p. 44) and Section 2.4.1 (p. 80).

Identification (equivalent to *browse* (Brehmer and Munzner, 2013) and *direct lookup* (Andrienko and Andrienko, 2006)) retrieves the characteristics of single objects (elementary, e.g., to read the data value of an object) or sets of objects (synoptic, e.g., to identify clusters).

Localization (equivalent to *locate* (Brehmer and Munzner, 2013) and *indirect lookup* (Andrienko and Andrienko, 2006)) retrieves the locations of target objects (elementary, e.g., search for objects with value 100) or target sets (synoptic, e.g., search for the members of class X).

Comparison (equivalent to *browse and compare targets* (Brehmer and Munzner, 2013) and *direct comparison* (Andrienko and Andrienko, 2006)) retrieves the differences in characteristics of objects (elementary, e.g., read the difference of two data values) or sets of objects (synoptic, e.g., how similar are the clusters/classes).

Relation-Seeking (equivalent to *locate and compare targets* (Brehmer and Munzner, 2013) and *relation seeking* (Andrienko and Andrienko, 2006)) retrieves the location of target relations of objects (elementary e.g., search for the objects that have a difference of 50 in attribute X) or sets of objects (synoptic e.g., search for the clusters that are most similar).

2

Design of Effective Color Encodings

Contents

This chapter is based on the publications [10, 8, 1]. The applications and use cases are based on [9, 16, 21]. Major parts of the sections appeared in the following publications:

[10] Sebastian Mittelstädt, Dominik Jäckle, Florian Stoffel, and Daniel A. Keim. *ColorCAT: Guided Design of Colormaps for Combined Analysis Tasks*. In Proceedings of the Eurographics Conference on Visualization, pages 115–119. The Eurographics Association, 2015.
[8] Sebastian Mittelstädt, Jürgen Bernard, Tobias Schreck, Martin Steiger, Jörn Kohlhammer, and Daniel A. Keim. *Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis*. In Proceedings of the Eurographics Conference on Visualization, pages 91–95, The Eurographics Association, 2014.
[1] Jürgen Bernard, Martin Steiger, Sebastian Mittelstädt, Simon Thum, Daniel A. Keim, and Jörn Kohlhammer. *A survey and task-based quality assessment of static 2d color maps*. In SPIE 9397, Visualization and Data Analysis, DOI:10.1117/12.2079841, 2015.

For the division of responsibilities and work, as well as a statement of contributions in these publications, please refer to Section 1.5 (p. 18).
The following contributions go beyond the published work:

1. An analysis of the visual channels hue, saturation, and lightness with respect to *pre-attentively* and *faithfully* encoding nominal, ordinal, and quantitative information.
2. Quality metrics and guidelines that measure the *pre-attentiveness* of colormaps for elementary analysis tasks.
3. Quality metrics that consider the background of visualization for color encoding.
4. Heuristics for initializing the optimization as well as to estimate the intersection and volumes of convex hulls for projecting high-dimensional data to perceptual uniform color spaces for effectively encoding data relations.
5. Perceptually optimized methods for encoding data relations for color blind persons.

Chapter Abtract & Structure

Color is used in almost every data visualization to encode information. In most applications, however, colormaps are selected without reasonable justification. As there are several pitfalls in colormap design and sources of biases, this leads to artificial interpretations of data.

For example, the rainbow  colormap tends to segment continuous data and let the analyst perceive artificial visual structures. Further, it hides high-frequency information and steers the attention of the analyst towards some arbitrary data values. Yet, this colormap is probably the most frequently used colormap in the history of information visualization and the question is why? However, "why" a specific colormap is selected for a specific data visualization is a very tough question for most visualization experts to answer.

This chapter answers the question "why" in several ways. First, it answers "why" some colormaps bias the analyst in one application but perform very well in another application. For example, even the rainbow colormap is very good for certain analysis tasks: In the task of reading (identifying) metric quantities, users are significantly (eight times) more accurate than with colormaps that are proposed by the state-of-the-art to encode quantitative attributes. Second, this chapter shows how to design colormaps for target applications based on the analysis tasks and how to justify "why" the colormap was selected or designed.

Effective color encodings are *faithfully*, *expressive*, and *pre-attentively* perceived, *support the analysis task* of the target application and are *semantic consistent*. This chapter aims to cover all the aspects and provides methods and guidelines for the effective color encoding of single data dimensions and high-dimensional data relations. The first part of *faithfully* representing data can, however, only partially be satisfied because this chapter has the focus on the design of color encodings. However, perceptual effects such as simultaneous contrast occur only in the rendered visualization and cannot be avoided *a priori* in the design of colormaps. These effects skew the *faithfulness* of colormaps but they can be compensated by the techniques in Chapter 3 and Chapter 4.

This chapter is organized as follows: Section 2.1 introduces the research challenge and defines the contribution. Related work on colormap guidelines and generation is discussed in Section 2.2. In Section 2.3, we provide guidelines and a tool for the guided design of color encodings for single data dimensions by supporting combined analysis tasks for real applications. Section 2.4 provides a requirement analysis and methods for the color encoding of (high-dimensional) data relations. Section 2.5 concludes this chapter and reveals open research gaps.

2.1 Challenges for Effective Color Encoding

The research of "how to encode data with color" has a long history in information visualization. There are opposing results and debates on how information can be encoded accurately with color resulting in different guidelines and colormaps. It is clear that the "optimal colormap", which is appropriate for all analysis tasks, does not exists (Rheingans, 2000). But there remains the main research question: how can we effectively apply color to encode information?

In our definition of effectiveness, color encodings have to *support the analysis task* (in the domain) of the user. Most of the existing guidelines and tools are data-driven (e.g., Brewer (2015) and Wijffelaars et al. (2008)). These guidelines provide colormaps that (in our definition) *pre-attentively* represent different data or attribute types. There exist also rule-based guidelines and tools (e.g., Bergman et al. (1995), Rheingans (2000)). They consider data- and *representation task* properties for their guidelines but they focus on single *elementary* analysis tasks. The elementary analysis tasks of data visualizations are *localization*, *identification*, and *comparison* of data values (see Section 1.6.2 (p. 27) for a detailed discussion). This was introduced for color mapping strategies by

Tominski et al. (2008), who focus rather on different data transformations than on designing colormaps.

The open challenge is that different tasks have different requirements for the visual encoding. For example, *comparing* data values requires that perceived distances match data distances. This is typically accomplished with unipolar colormaps that do not vary over different hues. These colormaps are the results of todays tools for continuous (sequential) data (Wijffelaars et al. (2008), Harrower and Brewer (2003), Bergman et al. (1995)). However, these colormaps are insufficient in the task of *identifying* data values (e.g., read metric quantities) because they do not provide many distinct colors (Ware, 1988). Color scales that are effective in *identification* must vary over multiple hues (Ware, 1988), but this distorts perceptual linearity and biases the analysts in the *comparison* task (Rogowitz et al., 1996). This is one of the main debates in colormap research since the colormaps that vary over multiple hues are considered as "rainbow colormaps", which "are known" to be harmful (Rogowitz et al., 1996). The complexity for designing colormaps increases if tasks are combined, e.g., to *identify* and *compare* data values, which is a typical task in real applications if the analyst aims to explore metric properties and patterns of the data. We argue, therefore, that the current guidelines are not sufficient, since real analysis tasks typically require the combination of different *elementary* tasks, especially if the goal is to explore the data. Another issue is that most approaches do not consider user preferences and culture that has high impact on the *semantic consistency* of the colormap and, thereby, on the acceptance in the target domain.

Further, the typologies of visual analysis tasks include several analysis tasks that are not covered yet, e.g., the *synoptic* tasks of Andrienko and Andrienko (2006). These tasks are performed on sets of objects. Thus, they require encoding (high-dimensional) data relations, which demand for more complex requirements. There is little work on perceptually preserving the relation of data objects that are obligatory for higher level analysis tasks, e.g., identify or compare clusters encoded with color.

These challenges sub-divide the requirement analysis and methods for color encoding into two levels: the encoding of (single) data dimensions for elemen-

tary tasks (Section 2.3 (p. 43)) and the encoding of (high-dimensional) data relations for synoptic tasks (Section 2.4 (p. 76)).

2.1.1 Contributions

This chapter provides the following contributions:

1. An analysis of the visual channels hue, saturation, and lightness with respect to *pre-attentively* and *faithfully* encoding nominal, ordinal and quantitative information. Further, their accuracy for absolute and relative judgments is discussed (Section 2.3.2 (p. 45)).

2. Perceptual quality metrics (Section 2.3.3 (p. 51)) and guidelines (Section 2.3.3 (p. 57)) for *effectively* encoding data dimensions with color. Further, effective colormaps for elementary analysis tasks and their combinations for real applications.

3. The tool *ColorCAT* that guides visualization designers through the creation of *effective* colormaps to provide *semantic consistent* colormaps by incorporating domain- and culture depended requirements as well as user preferences (Section 2.3.4 (p. 62)).

4. Data depended (perceptual) quality metrics for *effectively* encoding relations of (high-dimensional) objects with color (Section 2.4.2 (p. 81)).

5. An optimization method that projects relations of high-dimensional objects into perceptual uniform color spaces to provide *effective* color encodings for synoptic analysis tasks such as the perceptual identification of high-dimensional clusters within the data (Section 2.4.3 (p. 85)).

2.2 Related Work

2.2.1 Guidelines for Encoding Data Dimensions

The following paragraphs discuss research in guidelines and colormap algorithms for encoding single data attributes, as well as a controversy in the research field, in order to highlight and clarify the research goal of Section 2.3 (p. 43). Further, this section discusses the related work for encoding data relations, which is extended by the guidelines and algorithm provided in Section 2.4 (p. 76).

Guidelines for Encoding Data Attributes

The probably most famous colormap is the "rainbow" colormap that varies over the full spectrum of saturated colors. It is widely applied in many domain applications and is widely accepted as "temperature" scale for weather forecasts and the default colormap in many visualization toolkits (Borland and Taylor II, 2007). However, there are numerous of papers that show that this colormap is harmful. It segments unsegmented data or introduces perceptual borders in continuous data (Ware, 1988), has no perceptual order (Borland and Taylor II, 2007), is not color-blind safe (Light and Bartlein, 2004), is not perceptually linear (Bergman et al., 1995), draws the attention of the analysts to arbitrary data values (Rogowitz et al., 1996) because of non-uniform saturation and intensity (see 'Visual Attention' (p. 53)), and masks high-frequency details (Rogowitz et al., 1996).

These issue are caused predominantly by using multiple hues within a colormap, which leads to perceptual non-linearity. Some research argues that this

is the reason why such colormaps fail to faithfully represent data (Rogowitz et al., 1996). However, some researchers found that such multi-hue colormaps perform very well for some tasks (Ware (1988), Brewer (1997b)) and some research explicitly introduce algorithms (Pizer and Zimmerman (1983), Robertson and O'Callaghan (1986),Ware (1988), Levkowitz and Herman (1992), Keim (2000), Kindlmann et al. (2002)) to design such colormaps (see below). For example, Ware (1988) found that the rainbow colormap outperforms the perceptual linear grayscale in the task of reading single metric quantities from displays: While the participants made 20% errors with the grayscale, they only made 2.5% of errors with the rainbow colormap (see Section 2.3.2 (p. 45)). However, he also showed that the rainbow colormap biases the analyst in perceiving natural forms and shapes. A task in which the grayscale is very effective (see Section 2.3.2 (p. 45)). Ware (1988) further showed that (so called) spiral colormaps that vary over hues, however, are linearly increasing in lightness are a good compromise for perceiving natural forms and shapes, and for reading metric quantities from displays. This shows that the effectiveness of a colormap depends not only on its properties but also on the analysis task that is performed with this colormap. Section 2.3 (p. 43), therefore, discusses perceptual foundations and provides guidelines for effective color encoding based on the analysis tasks and the perception of the target user.

The elementary analysis tasks of data visualizations are *localization*, *identification*, and *comparison* of data values (Section 1.6.2 (p. 27)). This was introduced for color mapping strategies by Tominski et al. (2008), who focuses rather on different data transformations but also provides examples of colormaps for single analysis tasks (see Figure 2.1).

Rogowitz and Treinish (1993) integrated their guidelines for faithful color encoding into the PRAVDA architecture. These guidelines are formulated by Bergman et al. (1995) who analyze the data properties and representation tasks (see Figure 2.2). The authors explicitly focus on perceptual linear colormaps for isomorphic representation (comparison task) as well as on segmentation (identification task) and highlighting (localization task) for different data types. Figure 2.2 shows their guidelines with example colormaps produced by *Color-*

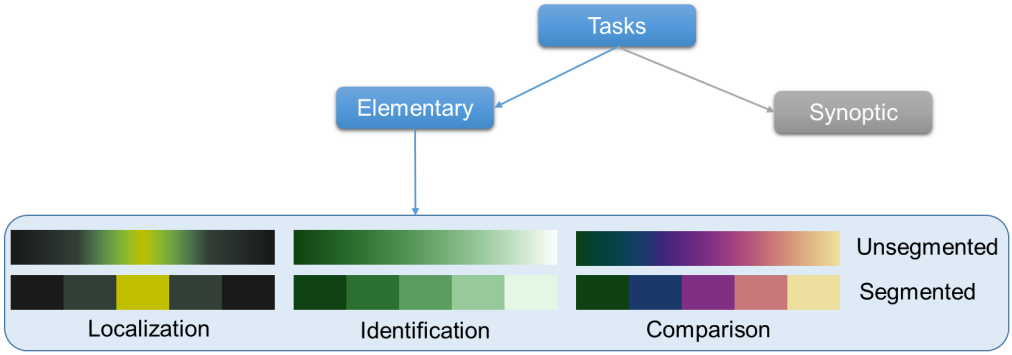


Figure 2.1: *Guidelines for color encoding for single elementary analysis tasks by Tominski et al. (2008). Colormap examples created with ColorCAT (Section 2.3.4 (p. 62)).*

CAT (Section 2.3.4 (p. 62)). Note, that their guidelines were extended to nominal and ordinal data by Rogowitz et al. (1996).

While elementary analysis tasks are well defined, real applications in which data visualizations are most effective, are often confronted with ill-defined (exploratory) tasks. Therefore, we argue that supporting single analysis tasks is not enough because real analysis tasks typically require the combination of different elementary tasks.

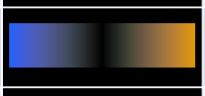
Data Type	Spatial Frequency	Representation Task			
		Examples	Isomorphic	Segmentation	Highlighting
Ratio (zero crossing)	Low		Luminance: Uniform Hue: complement Saturation: increasing	Many segments	Large range
	High		Luminance: increasing Hue: complement Saturation: increasing	Few segments	Small range
Interval	Low		Luminance: uniform Hue: opponent pairs Saturation: increasing	Many segments	Large range
	High		Luminance: increasing Hue: uniform Saturation: decreasing	Few segments	Small range

Figure 2.2: *Guidelines for color encoding by Bergman et al. (1995) for single representation tasks. Colormap examples created with ColorCAT (Section 2.3.4 (p. 62)).*

Cynthia Brewer provides guidelines (Brewer (1994), Brewer (1996a)) for color encodings predominantly for cartographic applications, that consider human perception to faithfully represent different data types (bivariate, diverging, sequential, and nominal data). Her color scales are handcrafted according to perceptual foundations. Further, the color scales provide perceptual linearity, avoidance of attention steering, color blind-safe-, and print-safe color encodings, which reduce also the impact of contrast effects. These guidelines and colormaps were made publicly available in the tool ColorBrewer by Harrower and Brewer (2003) (see Figure 2.3). The shortcoming of ColorBrewer according to our definition of effectiveness is that the colormaps are not *expressive* (provides only up to 12 colors) and do not support specific analysis tasks.

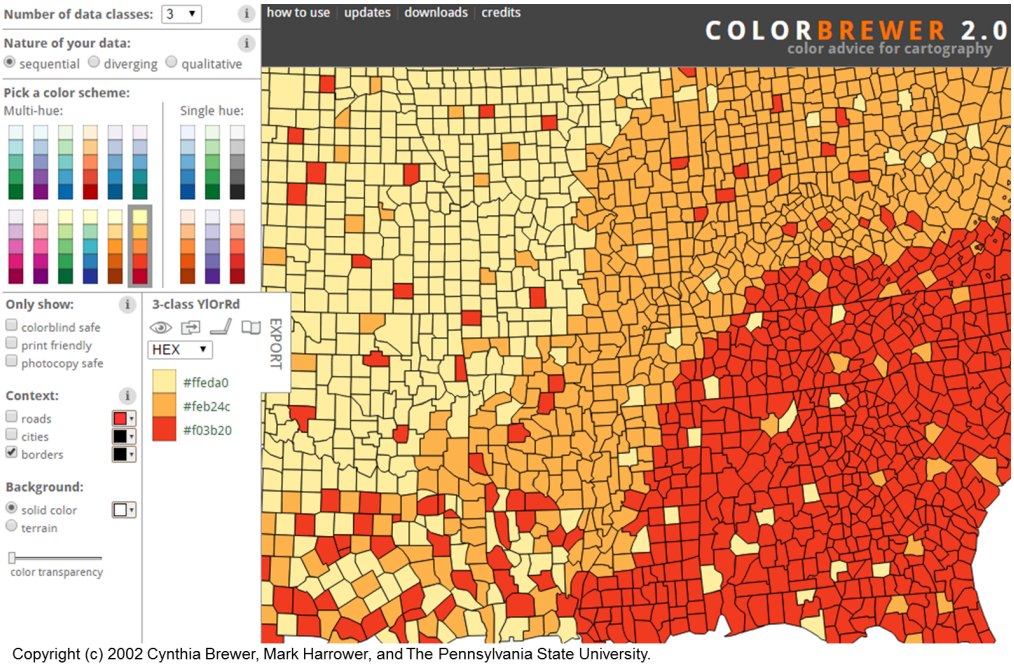


Figure 2.3: *Screenshot of ColorBrewer by Cynthia Brewer, Mark Harrower, and The Pennsylvania State University (2015). Retrieved June 04, 2015, from http://colorbrewer2.org/ .*

2.2.2 Colormap Generation

A survey of colormap guidelines and applied colormap research is presented by Silva et al. (2011), but this section focuses on related details to our algorithms. Requirements for colormap generation were introduced by Trumbo (1981), Pizer and Zimmerman (1983), and Levkowitz and Herman (1992). All highlight that colormaps must be perceptual ordered and that data distances must match perceptual distances (perceptual uniformity or linearity). The rules of Trumbo (1981) are used by Robertson and O'Callaghan (1986) to provide univariate and bivariate colormaps that vary perceptually linear over hues and lightness. Levkowitz and Herman (1992) define quality metrics for perceptual order, perceptual linearity, and color space exploitation (such that the colormap provides as many colors as possible). They formulate color mapping as an optimization problem and maximize color space exploitation (by varying over hues) while preserving perceptual order and perceptual linearity by varying over lightness. Rogowitz and Kalvin (2001) propose a perceptual method that allows end-users to efficiently evaluate and select color scales by assessing luminance variations in human faces. Kindlmann et al. (2002) exploit this technique to interact with the end-user to adapt colormaps to uncalibrated displays and different environment conditions such as ambient light. Wang et al. (2008) propose guidelines for the design of effective color palettes, from both aesthetic and attention-guiding points of view. They propose 13 rules for colormap design and provide algorithms for enhancing vividness and lightness as well as a framework for mixing colors in data visualizations. Wijffelaars et al. (2008) generalized the handcrafted colormaps of ColorBrewer and transfered their guidelines in an interactive tool that allows the design and customization of colormaps. Moreland provides a similar approach for diverging colormaps (Moreland, 2009).

Healey (1996) proposes a method for selecting colors that are effective in discriminating color encoded categorical data. The distance between colors is maximized in the perceptual linear color space CIELUV and equidistant colors are selected that share equal saturation and lightness, and avoid attention steer-

ing effects. Lee et al. (2012) provide a method that perceptually optimizes the visibility of categories in visualizations by balancing the saliency of the colors regarding the visual area of categories in a visualization. Lin et al. (2013) provide a method to select categorical colors from a set of colors according to the semantics of the data. The topic of each category is extracted and the method automatically identifies the color, which is probably semantically close to the topic by querying the web for images related to this topic. The color of the result images are extracted and the algorithm assigns the color for the category by expectation maximization.

All the methods are able to create well designed color encodings. However, they do not consider the target analysis task, target domain, and user preferences. Therefore, they are not able to sufficiently *support the analysis tasks* and cannot ensure *semantic consistent* colormaps.

2.2.3 Guidelines for Encoding Data Relations

Ware and Beatty (1988) performed an experiment, in which five dimensional data was mapped to two spatial and three color dimensions. The results indicated that each additional color dimension is as useful as an additional spatial dimension for cluster identification. Other guidelines of Brewer (1996a) and Ware (2012) suggest mapping two dimensions to hue and saturation (or lightness). This results in few distinguishable colors, which is in most cases enough to visualize effective overviews, however, it is hard to intuitively reconstruct a color mixture into two or three different dimensions without extensive knowledge about color perception (Ware, 2012).

Bivariate color schemes that meet several perceptual issues are discussed by Brewer (1996a) and Harrower and Brewer (2003). These schemes, however, do only support a limited number of color levels. An extension to this approach is introduced by Guo et al. (2005) and Guo et al. (2006). The method uses interaction and bell shaped rasters in the CIELAB space to produce diverging colors.

There is evidence that two-dimensional color maps are unintelligible for encoding certain dimensions (Wainer and Francolini, 1980). However, under a different perspective of visualizing the similarity of data points or clusters, these color maps have shown their usefulness in many papers. For example, Himberg (2000) and Bremm et al. (2011) project and scale high dimensional data to a lower (two) dimensional space to fit a two dimensional color map. Most methods interpolate in RGB or CIELAB between fixed color anchors in the corners. Some methods also use uniform planes of CIELAB (Wood and Dykes, 2008). Bernard et al. [1] surveyed visualization research papers containing two-dimensional colormaps and measured their quality according to the requirements for analysis tasks contributed in this thesis.

2.3 Color Encoding for Single Data Dimensions

Colormap design is challenging because the encoding must match the requirements of data and analysis tasks as well as the perception of the target user. A number of well-known tools exist to support the design of colormaps. ColorBrewer (Harrower and Brewer, 2003), for example, is a great resource to select colors. PRAVDAColor (Bergman et al., 1995) and Tominski et al. (2008), for example, provide valuable guidelines for single analysis tasks such as localization, identification, and comparison. However, for solving real world problems in most practical applications, single elementary analysis tasks are not sufficient but need to be combined especially for enabling exploratory tasks. There exist algorithms for sophisticated colormaps that may cover single task combinations. However, these algorithms are based on complex color spaces and optimization problems. The colormap designer has no influence on the outcome of optimized results. For instance, the colormap may lack in aesthetics (Wang et al., 2008)

but also may not be in-line with the mental model of domain experts (not *semantic consistent*) since the ordering of colors depends on culture and domain. This results in inappropriate colormap selection since there is no available tool that supports designers in the creation of colormaps for their analysis task. Further, colormaps for color-blind persons require additional strategies (Machado and Oliveira (2010), Sajadi et al. (2013), Oliveira (2013)) or recoloring methods (Kuhn et al., 2008).

In this section, we propose a methodology and tool to design colormaps for combined analysis tasks. We define color mapping requirements and develop a set of design guidelines. The visualization expert is integrated in the design process to enhance the *semantic consistency* of the colormap by incorporating his/her design requirements, which may depend on the application, culture, and aesthetics. Our *ColorCAT* tool guides novice and expert designers through the creation of colormaps and allows the exploration of the design space of color mapping for combined analysis tasks.

2.3.1 Requirements for Elementary Analysis Tasks

In the following, we define requirements for colormaps that encode the characteristics of objects according to single dimensions of objects. Since effective color encodings must support the analysis task of the user, the definitions are based on the elementary analysis tasks of data visualizations, where the focus is to identify and compare characteristics or to localize relations and characteristics of objects.

ER1 **Elementary Localization** retrieves the locations of target objects, e.g., search for objects with value 100 in attribute X. This requires that the visual references of objects that comprise the target characteristics stand out in the visualization and attract the visual attention of the analyst. The analyst should, therefore, be able to specify the characteristics of interest and the visualization should reveal “where” these characteristics can be

found. An ideal color encoding, therefore, highlights the target references on the display.

ER2 **Elementary Identification** retrieves the characteristics or the identity of single objects, e.g., to read an attribute value or all attributes of an object with a known location. With an ideal color encoding, the user can identify the absolute value by perceiving the color of an object. A good colormap, therefore, provides n distinct and salient colors for n distinct values of an attribute. The analyst looks up the color of objects in the color legend and reads the according data values. The requirement is, therefore, that colormaps provide as many colors as possible that are distinct with respect to human perception. In contrast to localization, all objects should be visually equal important in elementary identification. Otherwise, attention may be dragged towards single objects with higher visual importance and suppress other objects, which biases the analyst if the task is of exploratory nature.

ER3 **Elementary Comparison** defines the task of comparing the characteristics of two or more objects. Here, the analyst compares the colors of objects and decides whether they are similar or different. Furthermore, the analyst estimates the relative or absolute difference between the objects. The requirement here is that the data relations are preserved in the visual mappings. In other words, the distance of the objects in data space should be equivalent to the perceived color distance in the visualization space. Therefore, a conforming colormap has to be perceptually linear.

2.3.2 Perceptual Foundations for *Pre-Attentive* and *Faithful* Color Encoding of Data Attributes

While the last section discusses the requirements to support analysis tasks, this section analyzes how colormaps can be designed for *pre-attentive* perception

of data properties. If *pre-attentive* processes are exploited, the analyst can read visualizations more efficiently because he/she can simply perceive the information and does not need cognitive processes to understand the data. Further, this section provides a discussion of how accurate the human visual system is in making absolute and relative judgments.

From the opponent-process theory we can assume that there are three primary visual channels that let us perceive color: hue, lightness, and saturation (Palmer, 1999). The following paragraphs discuss the properties of these channels and highlight how they can be exploited for *faithful* and *pre-attentive* encoding of information.

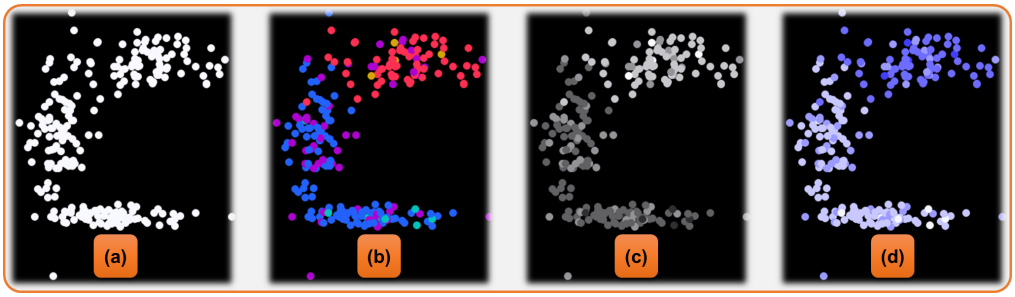


Figure 2.4: *Examples for categorical perception.* ***(a)*** *Scatterplot with arbitrary data without color encoding. Try to estimate the five color encoded categories in the other visualizations.* ***(b)*** *Our perception is efficiently perceiving all five categories encoded with different color hues (green, blue, purple, red, and orange).* ***(c)*** *and* ***(d)*** *show the same data encoded with lightness and saturation. However, our perception does not separate the categories effectively and we must browse the visualization in detail.*

Hue.

"Attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them" (Fairchild, 2013).

Color (hue) is effectively used to encode categories because chromatic vision evolved or is learned to discriminate between objects (e.g., bananas are yellow, cherries are red) (Harnad, 2003). Thus, it is easy for us to discriminate between objects because it is pre-attentively processed. For example, Figure 2.4(a) shows an arbitrary data set with a scatterplot and two dimensions encoded by

the X-axis and the Y-axis. In Figure 2.4(b) color is added to encode five categories. Since there are only few categories, it is intuitive for us to discriminate between the categories. **We conclude, that the chromatic channel hue *pre-attentively* supports *identification* of categories.** Of course, this has a limit in spatial frequency and the number of categories, but for simplicity we leave the statement for now (Ware (2012) provides a detailed discussion). Further, there are effects that interfere with categorical perception such as verbal interference (Roberson and Davidoff, 2000) but also the experience and culture of a person (Harnad, 2003). Therefore, categorical color encodings should adapt to user domain and preferences to be *semantic consistent*. In Figure 2.5(a) we use color hue to encode a quantitative attribute. Can we compare these colors? Is red 'more' than blue? Probably, but what about yellow and green? Studies showed that people tend to order color hues differently (Ware, 2012). **This shows that we cannot *faithfully* use hue to express ranks, ordinal-, or quantitative attributes since we are not able to perceptually *compare* color hues in a metric sense.**

Brightness & Lightness.

"Brightness: Attribute of a visual sensation according to which an area appears to emit more or less light.
Lightness: The brightness of an area judged relative to the brightness of a similarly illuminated area that appears to be white or highly transmitting" (Fairchild, 2013).

The achromatic lightness channel is very effective to detect natural forms, movements, and stereoscopic depth by perceiving the 'amount' of light from the light emitting or reflecting objects (Ware, 1988). For perceiving lightness or brightness, our eye relates the incoming light to its surround (Palmer, 1999) in order to detect edges for estimating the form that we recognize and classify. Thus, it allows us to discriminate between 'more' or 'less' light, which provides a channel for ordinal and quantitative information (see Figure 2.5(b)). However, it is hard for us to detect the absolute amount of light because our perception of lightness is relative (Palmer, 1999). Thus, it is also hard for us

to identify absolute metric quantities (see 'Absolute Judgements vs. Relative Judgements' below). **Hence, with lightness, we can *faithfully compare* the color encoded data values and perceive *pre-attentively* relative differences as well as ranks/orders in the data**. But if lightness is used to encode categorical data such as in Figure 2.4(c), it can be experienced that we have difficulties to discriminate between the categories in comparison to Figure 2.4(b). **Concluding, lightness is not effective in identifying categories and absolute differences of light**.

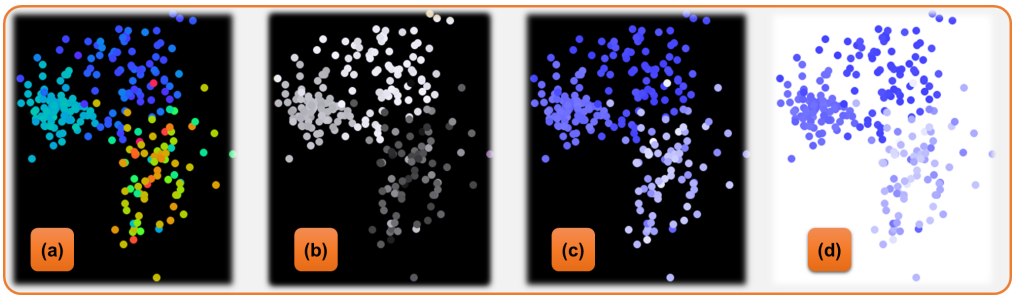


Figure 2.5: *Examples for metric perception. Try to localize the data points with the highest values.* ***(a)*** *Color hues do not have a natural order and we cannot solve the task without a color legend.* ***(b)*** *We can immediately spot the highest values with lightness and also compare the data values.* ***(c)*** *This is also possible with saturation. However, on black backgrounds, we are missing a reference to order colors with saturation pre-attentively (see Section 2.3.3 for a detailed discussion).* ***(d)*** *The white background is a reference for our perception to order colors according their saturation.*

Saturation.

"Colorfulness: Attribute of a visual sensation according to which the perceived color of an area appears to be more or less chromatic.
Saturation: Colorfulness of an area judged in proportion to its brightness" (Fairchild, 2013).

The second chromatic channel saturation let us perceive the vividness of a color (Palmer, 1999). The colors along the neutral grayscale (black-gray-white) are not saturated, which are mixtures of all wavelengths (of varying intensity). If the perceived light does only contain a single wavelength it appears fully saturated or vivid such as pure red, green, or blue light. Thus, the sensation of saturation is a measure of the pureness of wavelengths (Palmer, 1999). **This implies**

that saturation is another channel for ordinal and quantitative information. However, the range of saturation is limited. The number of discernible steps in saturation (approximately 20 just-noticeable-differences, JNDs) is only the half of the JNDs along the lightness range. Therefore, **saturation is not as effective as lightness in comparing color encoded quantities and has the same limitations to encode categorical information** (see Figure 2.4(d) and Figure 2.5(d)).

Absolute Judgements vs. Relative Judgements

The *relative judgment model* of Stewart et al. (2005) proposes that absolute judgments are achieved by a sequence of relative judgments, which implies that our perception does not process the absolute quantity of unidimensional stimuli. This approves findings of Eriksen and Hake (1955), who found that the human visual system has a limited channel capacity for absolute judgments with color including hue and brightness (lightness). They measured that the information transmitted in bits over each channel without error is $C = 2.34$ bits (channel capacity) for brightness and $C = 3.06$ bits for hue (their measure is equivalent to the Shannon-Weaver-Model expressed in Eq. 2.1 with bandwidth $B = 1$ and omitted 1 in the logarithm (Garner and Hake, 1951)). In this definition of channel capacity, the values for C can be used to calculate the signal-to-noise ratio $\frac{S}{N}$ for each channel (Eq. 2.2). Thus, we can identify 5 brightness levels ($\frac{S}{N} \approx 5.06$) and around 8 hue levels ($\frac{S}{N} \approx 8.45$) without error, according to Eriksen and Hake (1955). The noise-to-signal ratio indicates the influence of white noise within each channel. In other words, the participants made an error of around 19% if quantitative information is transmitted by the brightness channel ($\approx 5.06^{-1} * 100\%$) and around 12% with the hue channel ($\approx 8.45^{-1} * 100\%$).

$$C = log_2\left(\frac{S}{N}\right) \quad \text{original (Shannon and Weaver, 1949): } C = B \cdot log_2\left(1 + \frac{S}{N}\right) \tag{2.1}$$

$$\frac{S}{N} = 2^C \quad \text{original (Shannon and Weaver, 1949): } \frac{S}{N} = 2^{\frac{C}{B}} - 1 \tag{2.2}$$

In the experiment of Ware (1988), participants were shown a data visualization and a color legend. The task was to identify data values that were high-

lighted in the visualization according to the color legend (see *elementary identification* in Section 2.3.1 (p. 44)). For a grayscale colormap (brightness channel) Ware found that participants made an error of 17-20%, which confirms the findings of Eriksen and Hake (1955). Ware found that this is predominantly due to contrast effects. For the rainbow color scale (hue channel), Ware found that participants were significantly more accurate. They made only 2.5% of errors, which implies that participants could theoretically identify up to 40 levels of color hues without error ($\frac{N}{S} = \frac{2.5\%}{100\%} \mapsto \frac{S}{N} = 40$) if the color legend is provided, in contrast to the results of Eriksen and Hake (1955). This is further supported by measurements in perceptual uniform color spaces, since there are around twice as many discernible colors along the chromatic channel than along the lightness or saturation channel, which are required for the elementary identification task (Ware, 2012). This is an interesting finding, since the only difference between the experiments of Ware (1988) and Eriksen and Hake (1955) is that the participants could look-up the color in the legend; however, this seems to have only effect on the hue channel and not on the brightness channel.

Following the findings above, a color legend is required to encode information with color hues. The reason for this is the issue that there is no natural order of hues. If people are asked to order color hues, e.g., red, green, yellow, and blue, they will certainly disagree on the results of other people (Ware, 2012).

This issue makes a color legend obligatory and indicates that a look-up is required, which implies that absolute judgments with a color legend are not *pre-attentive* (Section 3.4.3 (p. 133)). This also shows if identification is supported by a colormap and legend, which exploits the hue channel, is far more accurate and *faithful* than the perception of absolute judgments with brightness (with or without a legend). However, this also implies that relative judgments along the hue channel should be avoided since a comparison would require two (cognitive) look-ups to determine a difference or to learn the color legend by heart. Due to the issue of natural unordered hues, the analyst may be biased in relative judgments. For relative judgments, the brightness and saturation channels are more *effective* since we can process relative judgments *pre-attentively*.

2.3.3 Quality Metrics and Guidelines for Effective Color Encoding

In this section, we define perceptual quality metrics and guidelines for effective colormaps. This covers the *support for (combined) elementary tasks* with *pre-attentive*, *faithful*, and *expressive* colormaps that make use of the perceptual foundations discussed in the last section.

Thereby, the guidelines can also generate color encodings for categorical and ordinal data (see Figure 2.13 (p. 63)) since the requirements of Section 2.3.1 (p. 44) are valid for these data types as well. The only difference that exist is that categorical data can only be *identified* or *localized* since there exist no absolute or relative differences between categories.

Quality Metrics

The following three paragraphs will introduce novel quality metrics for the elementary tasks *localization*, *identification*, and *comparison*. These require that the encoded data is *faithfully* perceived and the encoding is *pre-attentive* and *expressive*. The metrics are defined as cost functions such that minimizing these functions provides high qualitative color encodings.

The background is of major importance for effective color encoding because color appearance is relative. Thus, the background can be exploited to become a perceptual reference for our metric encodings (see $QM3$). However, it can also introduce contrast effects (see Chapter 3), and further limit the visual channels, e.g., it is much easier to perceive variations of lightness on black backgrounds than on white backgrounds, because “white” already saturates our cones (see $QM1$ and $QM3$).

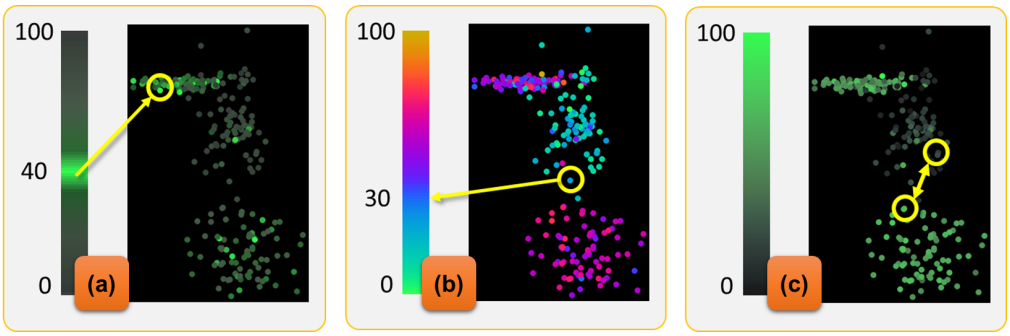


Figure 2.6: *Elementary analysis tasks:* ***(a)*** *Localization: search for specific (known) values on the display;* ***(b)*** *Identification: read the (unknown) value from a color encoded object;* ***(c)*** *compare two or more color encoded objects to read/perceive their absolute or relative difference.*

① QM1-Localization.

This task is performed when the analysts wants to see "where" objects with a specific value are *located* within the data (Section 1.6.2 (p. 27)), e.g., *visual query for the value 100 on the display* (see Figure 2.6(a)). Therefore, data values and ranges of high importance must be perceptually striking in the visualization (e.g., highlighted). To provide a *faithful* and *expressive* color mapping, we propose that the visual importance VI of a color c_i must encode the data importance DI of value i and introduce Eq. 2.3. Studies showed that visual attention is predominantly steered by intensity and saturation (Camgöz et al., 2004). Thus, we approximate VI with the arc of intensity I and saturation S (in the HSI color space of Keim (2000)), which is in-line with the approach of Guo et al. (2005) and results of ColorBrewer (Harrower and Brewer, 2003). However, the *pre-attentiveness* of highlighting depends on the background (see Figure 2.7). If one visual channel is already saturated (e.g., intensity on white background) then variations in these channels will not have the required highlighting effect. The data values that should fade out, would still be visible (see Figure 2.7(b)). Therefore, we propose that the contrast between the values that should be highlighted and the background has to be maximized and further, the contrast between data values that should fade out and the background has to be

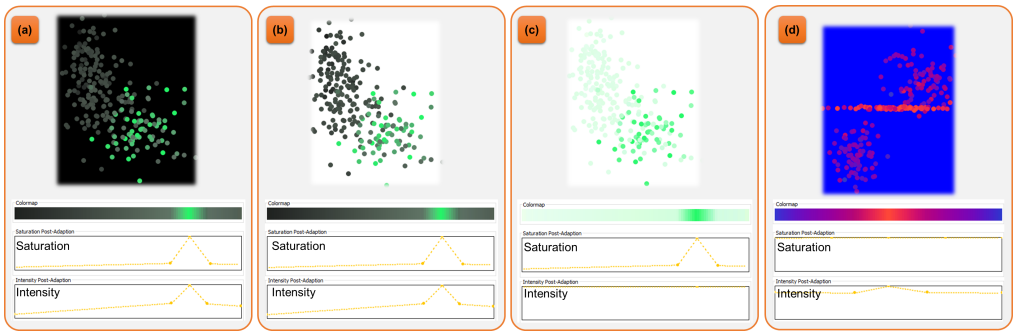


Figure 2.7: *Elementary localization on different backgrounds. Due to different backgrounds not all visual channels work equally well. While saturation and lightness work well on black backgrounds to highlight data values* ***(a)****, lightness does not work well on white backgrounds because the data values outside the selection appear salient to the analyst* ***(b)****. This is resolved by maximizing the intensity of the colors and exploiting saturation for highlighting* ***(c)****. On colored background, highlighting must exploit the variation of the hue channel while preserving the saturation and intensity of the background color. In* ***(d)*** *a gradient from blue to red highlights some data values (lightness is also used since blue is a dark color and the human eye has less blue cones than red and green cones).*

minimized (see Figure 2.7(c,d)). Therefore, we introduce Eq. 2.5 for estimating the visual importance of a color. We calculate $QM1$ (Eq. 2.3) with Eq. 2.5 for the visual channels hue, saturation, and intensity (e.g., with I_{ref} indicating the intensity of the background and $\alpha = 60°$). This is also valid for colored backgrounds, e.g., values can be highlighted on a saturated and intense blue background by a variation in hue (see Figure 2.7(d)).

$$min(QM_1) = min(\sum_i |DI(i) - VI(c_i)|) \tag{2.3}$$

$$VI_{attention}(c) = \sqrt{I_c^2 + S_c^2} \tag{2.4}$$

$$VI_{contrast}(c) = \sqrt{\Delta(H_c, H_{ref}) + (S_c - S_{ref})^2 + (I_c - I_{ref})^2} \tag{2.5}$$

$$\Delta(H_c, Href) = \begin{cases} 1, if|H_c - H_{ref}| > \alpha \\ 0, else. \end{cases}$$

② QM2-Identification.

This task is performed when the analyst *browses* or *explores* the data and reads values from color encoded objects on the screen (Section 1.6.2 (p. 27)), e.g., *estimate the value of the upper-left object* (see Figure 2.6(b)). The experiments of Eriksen and Hake (1955) and Ware (1988) showed that the chromatic channel hue is *pre-attentive* and *faithful* for *identification* (Section 2.3.2 (p. 49)). This has three reasons: 1) this visual channel has evolved to *identify* or discriminate objects (Harnad, 2003); 2) the impact of contrast effects is reduced in comparison to the brightness channel (Ware, 1988); 3) the chromatic channel hue provides twice as many perceptually distinct colors (JND, just-noticeable-difference (Mahy et al., 1994)) than the other visual channels lightness and saturation (Ware, 2012). Since, this thesis provides methods to compensate for contrast effects (Chapter 3), we focus here on (1) and (2).

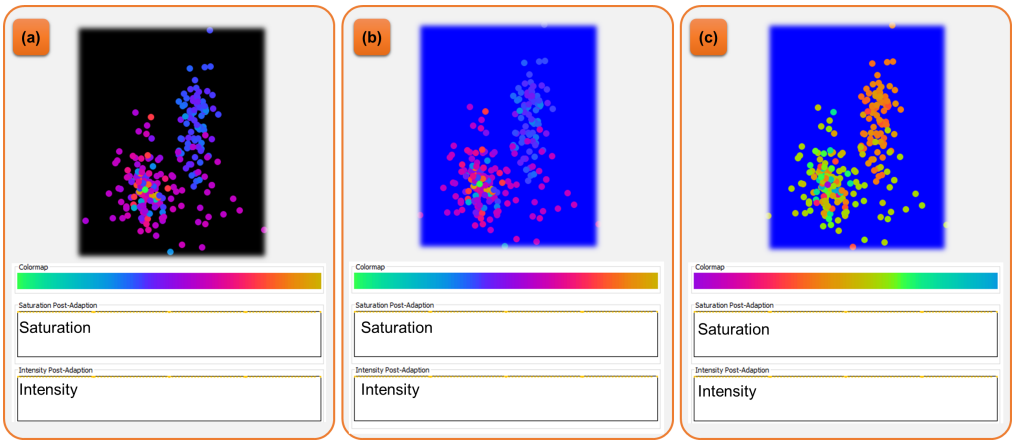


Figure 2.8: *Elementary identification on different backgrounds. In* ***(a)*** *all colors can be clearly identified. In* ***(b)*** *some "blue" data values are masked by the blue background. With a colormap that avoids the background color* ***(c)****, the data can be clearly identified since the color difference to the background allows perceptual discrimination.*

A high number of perceptually distinct colors (JNDs) allows, thereby, *faithful* and *expressive* identification of data values (Eq. 2.7). The *identification* task requires that the number of JNDs is maximized and all colors share equal visual importance (Eq. 2.8 with VI_{avg} being the average visual importance). This avoids the typical harmful effects of "rainbow" colormaps (Rogowitz et al., 1996)

such as attention steering effects or intensity gaps. In order to measure the amount of JNDs ($|\{JND_{CM}\}|$) of colormap CM, we propose to segment CM such that the colors c_i within each segment M are perceptually equal to the centroid of the segment ($\Delta E(c_i, c_M) \leq JND$), but distinct ($\Delta E(c_i, c_{M'}) > JND$) to the centroid of other segments M' (equivalent to ***MacAdam ellipses*** (MacAdam, 1942) in color spaces or within a 2D colormap [1]). The number of segments corresponds to the number of JNDs (Eq. 2.7).

$$min(QM_2) = min(f_{21}, f_{22}, f_{23}, f_{24}) \tag{2.6}$$

$$f_{21} := 1/|\{JND_{CM}\}| \tag{2.7}$$

$$f_{22} := \sum_i |VI(c_i) - VI_{avg}| \tag{2.8}$$

$$f_{23} := \sum_i 1 - S_i \tag{2.9}$$

$$f_{24} := \sum_i max(JND - \Delta E(c_i, c_{ref}), 0) \tag{2.10}$$

$$\Delta E := \sqrt{(\Delta J/K_L)^2 + \Delta a^2 + \Delta b^2} \text{ see Luo et al. (2006)} \tag{2.11}$$

$$\{JND_{CM}\} := \{M, M' \subset CM | \forall c_i \in M : \tag{2.12}$$
$$M \neq M' \wedge \Delta E(c_i, c_M) \leq JND \wedge \Delta E(c_i, c_{M'}) > JND\}$$

For *pre-attentive* "object identification", colormaps should vary over the hue channel with maximized saturation (Section 2.3.2 (p. 45)), which can be measured with f_{23} and S being the saturation of i (in combination with f_{21}).

Some colors could be potentially masked by the background (see Figure 2.8(b)). For backgrounds along the grayscale, this is not critical since colormaps for identification typically vary along the hue channel. However, for colored backgrounds, there must be at least one JND color difference between the background and the colors of the colormap such that the analyst can perceive the data objects as such (see Figure 2.8(c)). Therefore, we propose Eq. 2.10, which adds penalties to colors that are not differentiable from the background c_{ref}.

The goal to minimize the cost function QM_2 in order to support *identification* is, thereby, a multi-objective optimization problem (Eq. 2.6, see below for a solution).

③ QM3-Comparison

Elementary comparison is about comparing two or more visual encoded objects (Section 1.6.2 (p. 27)) and to perceive the relative and absolute differences (see Figure 2.6(c)). This task requires that distances in data space are equal to perceived distances in the visual encoding (Rogowitz et al., 1996). The color encoding is *faithful* if the color distances $\Delta E(c_1, c_2)$ of two data values reflect the data distance $d(d_1, d_2)$. Therefore, we propose to measure perceptual uniformity with the *Sammon's stress measure* (Eq. 2.14) by Sammon (1969). As described in Section 2.3.2 (p. 45), our perception and cognition is not able to order hues, while lightness and saturation provide *pre-attentive* perception of ordinal and metric information. In order to make use these effective channels, colormaps should vary predominantly over lightness and saturation, which further perceptually orders also hues (Ware, 1988).

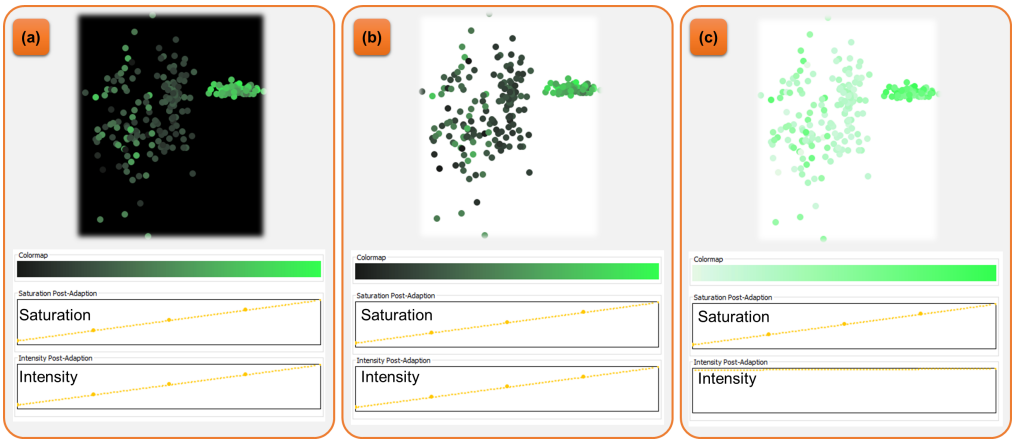


Figure 2.9: *Elementary comparison on different backgrounds. Due to different backgrounds not all visual channels work equally well. Saturation and lightness work well on black backgrounds to compare data values, because black is not intense and not saturated. Therefore, black is a good reference for comparison* ***(a)****. In contrast, lightness does not work well on white backgrounds because the background is, due to its intensity, not a pre-attentive reference for the analyst* ***(b)****. This is resolved by maximizing the intensity of the colors and exploiting saturation for ordering of the colors* ***(c)****.*

The background has also great impact on this task, since it limits the perception of variations if the visual channel is already saturated and it can be-

come a reference for the color encoded metric quantities (see Figure 2.9(a)). As the axis in a scatterplot, the background color ideally encodes the zero (or reference value) for estimating absolute or relative differences. Thus, on dark backgrounds, colormaps should vary perceptual linear from low intensity (low values) to high intensity (high values). Bright backgrounds excite already in intensity; thus, intensity is less pre-attentive to order colors (see Figure 2.9(b)). However, bright colors are often not saturated and colors can be ordered by saturation (see Figure 2.9(c)). Since perceptual uniform color spaces (e.g., CIECAM02-UCS) allow vector arithmetics, we propose to measure perceptual linearity by modeling the color vectors of a colormap with linear regression and estimating the error of the model (Eq. 2.15 with b being the background color, i being the index of color c_i, and gradient m being estimated by the model). In order to provide *expressive* color encodings for the comparison task, the number of distinct color levels has to be maximized (f_{21}). Colormaps that support the comparison task must, therefore, minimize f_{21}, f_{31}, and f_{32} (Eq. 2.13).

$$min(QM_3) = min(f_{21}, f_{31}, f_{32}) \tag{2.13}$$

$$f_{31} := \frac{1}{\sum_{i<j} d(d_i, d_j)} \sum_{i<j} \frac{(d(d_i, d_j) - \Delta E(c_i, c_j))^2}{d(d_i, d_j)} \tag{2.14}$$

$$f_{32} := \sum_i |c_i - m \cdot i - b| \tag{2.15}$$

Guidelines for Maximizing Quality

Requirements and quality metrics for elementary tasks can be precisely defined because the tasks are well-defined. However, real analysis tasks are often ill-defined; therefore, we aim to combine the elementary analysis tasks and provide guidelines for designing effective color encodings for real applications.

Note, that the quality metrics are defined as cost functions such that the cost functions $QM_1 - QM_3$ must be minimized to maximize the quality of the color encoding. Creating effective colormaps can, therefore, be formulated as a multi-

objective optimization problem with Eq. 2.16. The weights α may be adjusted according the combination of the elementary tasks.

$$min(\alpha_1 \cdot QM_1, \alpha_2 \cdot QM_2, \alpha_3 \cdot QM_3) \tag{2.16}$$

$$min(\sum_{i=1}^{3} \alpha_i \cdot QM_i) \tag{2.17}$$

The optimization goal Eq. 2.16 can be reached by a variety of optimizers. It is possible to formulate multi-objective optimization problems by a sum of all objectives (Eq. 2.17) and perform single-objective numerical optimization. Without any gradient, combinatorial optimizers and evolutionary algorithms could explore the full color space and find the optimal combination of colors that minimizes the cost functions. For higher efficiency, the gradient for minimizing Eq. 2.17 can be estimated to enable numerical optimization that efficiently converge to an optimum (Section 2.4.3 (p. 85) and Section 3.3.3 (p. 121) provide examples). However, optimization is typically a black box and often non-deterministic. Since general quality metrics are not able to cover *semantic consistency* depending on specific user preferences, culture, and application domain constraints, we aim to integrate the visualization designer into the algorithm. The following paragraphs discuss heuristic guidelines to minimize the quality metrics for (combined) elementary analysis tasks and provide means for effective colormaps. These guidelines are the base for the colormap algorithm of *ColorCAT* (Section 2.3.4 (p. 62)). Figure 2.10 shows examples created with *ColorCAT*. The encircled numbers link colormaps to the task combinations.

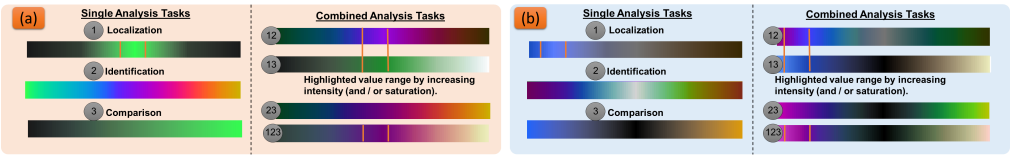
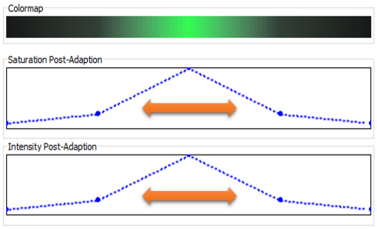
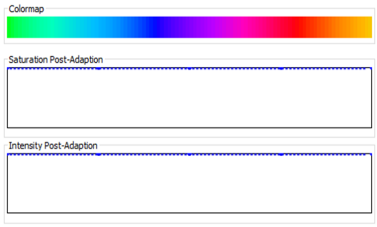


Figure 2.10: Image taken from [10]. *Colormaps for analysis tasks (① localization, ② identification, ③ comparison) and their combinations for sequential **(a)** and diverging **(b)** data. The colormaps vary perceptually linear over hue, saturation, and intensity according to the task combination. All colormaps presented here are color-blind safe besides ②, which maximize JNDs for normal color vision.*



① Localization.

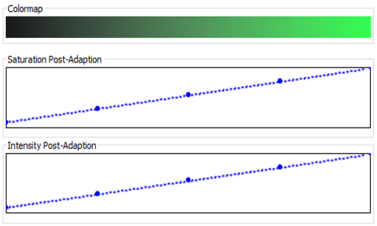
In most applications, this is an exploratory task that requires the input of the user. Therefore, the user must be able to specify the data importance. Then, QM_1 can be minimized by selecting one color hue and increasing in intensity and saturation according to the specified data importance, e.g., from black to green (sequential) or blue and orange (diverging, Figure 2.10(b)). *ColorCAT* lets the user specify the data importance by interactive spline charts and models intensity and saturation accordingly (see Figure 2.17 and Section 2.3.4). On black backgrounds, saturation and intensity should be used. However, on white backgrounds the attention is predominantly steered by saturation. Thus, intensity should be maximized and saturation must model the data importance for highlighting.



② Identification.

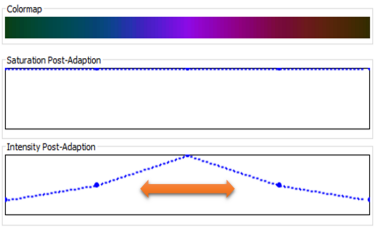
For minimizing QM_2, sequential colormaps must vary perceptually linear over the full range of hues with maximized but equalized intensity and saturation. An alternative to increase the number of distinct colors with equal visual importance is increasing intensity while decreasing saturation (Figure 2.11). This is recommended for color-blind persons for whom mixing red and green tones must be omitted. Accordingly, diverging colormaps increase in saturation but decrease in intensity.

Figure 2.11: Image taken from [10]. *Colorblind-safe colormaps for identification. Top sequential, bottom diverging.*



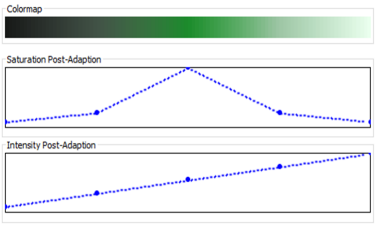
③ Comparison.

Perceptual uniformity and linearity is required for perceiving absolute differences ($QM3$). To provide perceptual uniform and linear colormaps for sequential and diverging data, the colormaps must vary from single color hues to black with perceptual linear decreasing saturation and intensity. However, perceiving absolute differences is anyway very limited by our perception (see Section 2.3.2 (p. 45)). Colormaps that vary over multiple hues and linear over intensity are perceptually ordered and thus, enable *pre-attentive* relative judgments and *faithful* absolute judgments (see also ㉓). Since relative judgments are more effectively supported by our perception, it makes sense to relax the requirements of perceptual uniformity and linearity and focus on the *pre-attentive* support for relative judgments.



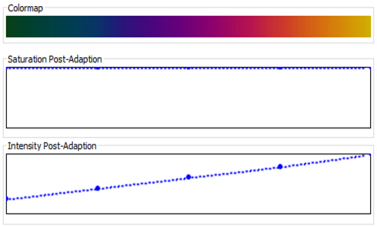
⑫ Elementary Exploration.

It is possible to build combinations of the different tasks, e.g., the analysts wants to *locate* specific objects and at the same time to *identify* (*browse*) the values of other objects (this supports the *exploration* task, see Section 1.6.2 (p. 27)). To support identification, the colormaps must vary over hues with a maximum of saturation ($QM2$). Intensity is increased to highlight the value ranges of interest ($QM1$).



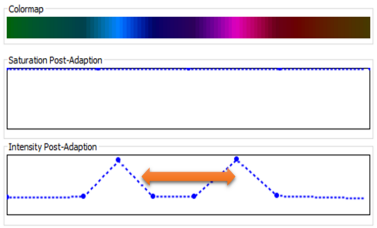
⑬ Comparative Exploration.

This combination comprises the *localization* and *comparison* of data values. Complete perceptual uniformity and linearity cannot be provided because some value ranges must stand out due to highlighting. Therefore, colors must be perceptually ordered by linear increasing intensity ($QM3$) and saturation must increase to highlight the specified data objects ($QM1$).



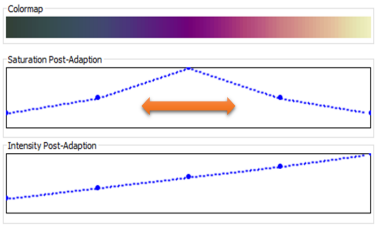
(23) Metric Exploration.

The most common analysis task combination is that the analyst wants to *identify* but also to *compare* data values. The challenge is to provide perceptual linearity and many distinct colors simultaneously. The results are the well-known *spiral colormaps* (Levkowitz and Herman (1992), Keim (2000), Ware (2012)) for sequential data that vary over hues with a maximum of saturation ($QM2$). The increasing intensity perceptually orders the colors, and thereby, reduces the bias of non-linearity on relative judgments ($QM3$). In Chapter 3 (p. 126), we provide experiments to validate this combination.



(12) Relation Seeking.

The analyst specifies a relation, for which the visual references should be highlighted, e.g., find all objects with a difference of 50. This task requires that the visual references are highlighted and the data values can be identified ($QM2$). Perceptual linearity cannot be preserved globally since the objects of interest must perceptually stand out. The user must be able to specify the relation of interest, which can be encoded by intensity ($QM1$).



(123) Trade-off.

The combination of all tasks is not recommended because this results in colormaps with many trade-offs. By varying over hues with linear increasing intensity ($QM3$) and a minimum of saturation ($QM2$), values can be *identified* and *compared*. However, to support *localization*, saturation needs to be increased to the specified value range ($QM1$).

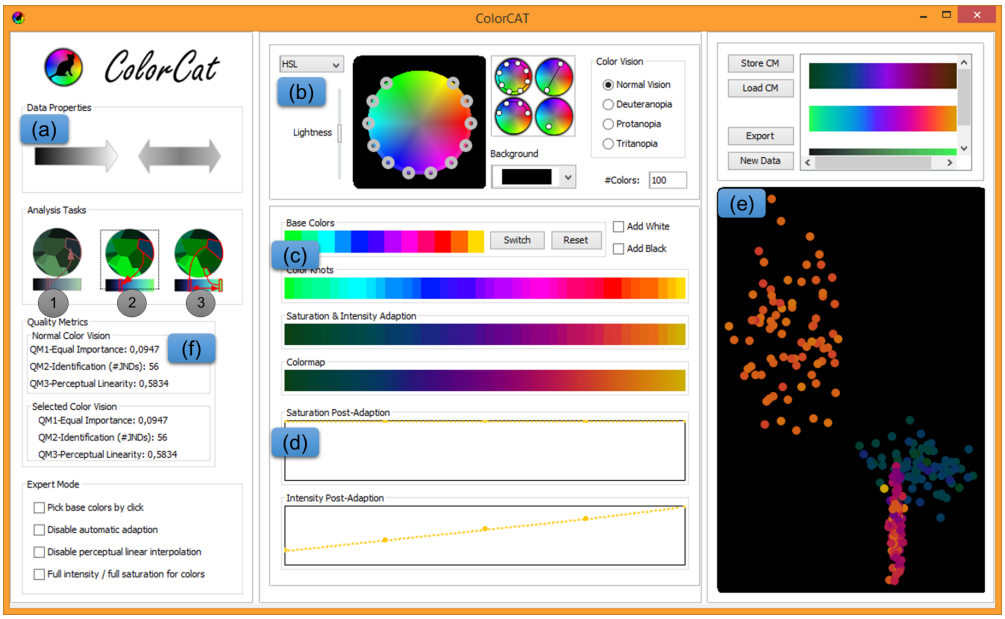


Figure 2.12: Image taken from [10]. *Screenshot of ColorCAT.* ***(a)*** *The user selects the data type and analysis task combination (① Localization, ② Identification, ③ Comparison) and ColorCAT suggests a colormap design based on the task requirements (see Figure 2.10 (p. 58)).* ***(b)*** *The user can select different base colors in the color picker, which allows free adding, removing and rotating knots.* ***(c)*** *visualizes the steps of the colormap algorithm for the designer to understand how the base colors are modified to match the analysis task.* ***(d)*** *Splines allow advanced users to modify intensity and saturation of colormaps.* ***(e)*** *The scatterplot allows visual inspection of colormaps.* ***(f)*** *Quality metrics reveals the quality of colormaps. Each element is discussed and illustrated in the following.*

2.3.4 *ColorCAT*: Interactive Guided Design of Effective Colormaps

The idea of *ColorCAT* (Design of Colormaps for Combined Analysis Task) is that the designer specifies the properties of the data (sequential, diverging) and selects the analysis task combination (see Figure 2.12). *ColorCAT* then derives the requirements (ER1, ER2, and ER3) for the selected tasks and models the intensity and saturation gradient of the colormap to minimize the quality met-

rics $QM1 - QM3$. The designer selects and orders base colors for *semantic consistency* especially for *identification* tasks to provide multiple hues (QM2). The colormap is generated by interpolating between the base colors in a perceptual uniform color space to maximize *expressiveness* (QM2) and to maximize *faithfulness* by perceptual linearity and/or perceptual order (QM3). Advanced designers are able to interactively change all properties of the colormap.

Support of Attribute Types.

All colormaps in this section are designed to map quantitative data. *ColorCAT* can also generate color encodings for categorical and ordinal data (see Figure 2.13) since the requirements of Section 2.3.1 (p. 44) are valid for these data types as well. The quality metrics and guidelines are based on the perceptual foundations of Section 2.3.2 (p. 45) and thereby also capture ordinal and nominal data. For example, the user can specify the number of colors and the colormap algorithm provides a set of colors for encoding categories (Figure 2.13 (p. 63)). The difference is that categories can only be *identified* or *localized* since there exist no absolute or relative differences between categories.

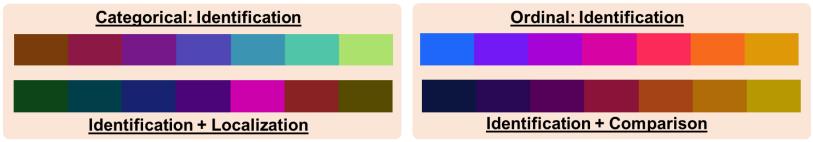


Figure 2.13: Image taken from [10]. *Examples for categorical and ordinal colors.*

Interactive Selection of Base Colors.

ColorCAT provides an interactive color picker that visualizes the intuitive HSL color space. *ColorCAT* supports the user by suggesting color orderings for, e.g., *spiral colormaps* and color harmonies for diverging colormaps. In this way, the

designer can order the colors according to domain, culture, or user preferences and thereby enhance the *semantic consistency* of colormaps. Expert users can scan through the HSI and CIECAM02 color spaces and create customized colormaps beyond our guidelines.

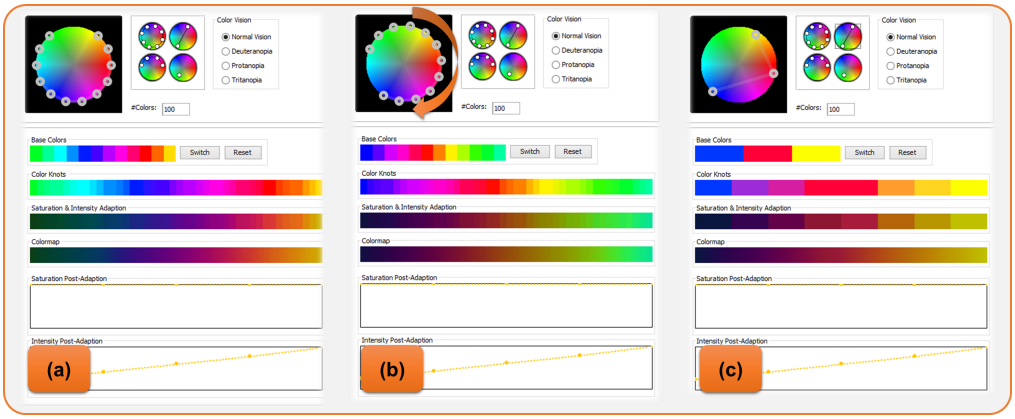


Figure 2.14: *Interactive selection of base colors. The base colors can be adapted by rotating the knots, e.g., from (a) to (b). (c) The knots can also be placed on specific colors.*

Color Vision Deficiencies

In order to design colormaps that are color-blind safe, *ColorCAT* can be switched into three types of color blindness: protanopia, deuteranopia (most common), and tritanopia. All colors in *ColorCAT* will be simulated by the approach of Brettel et al. (1997) according to the selected deficiency and thus, colormap designers can perceive how the colormap will appear to a color-blind person (see Figure 2.15). Additionally, *ColorCAT* indicates the quality by $QM1 - QM3$ of the colormaps for the selected color blindness. This mainly influences the base color selection since mixing red and green hues should be avoided. However, also these hues can be mixed if the difference in intensity between these hues is high.

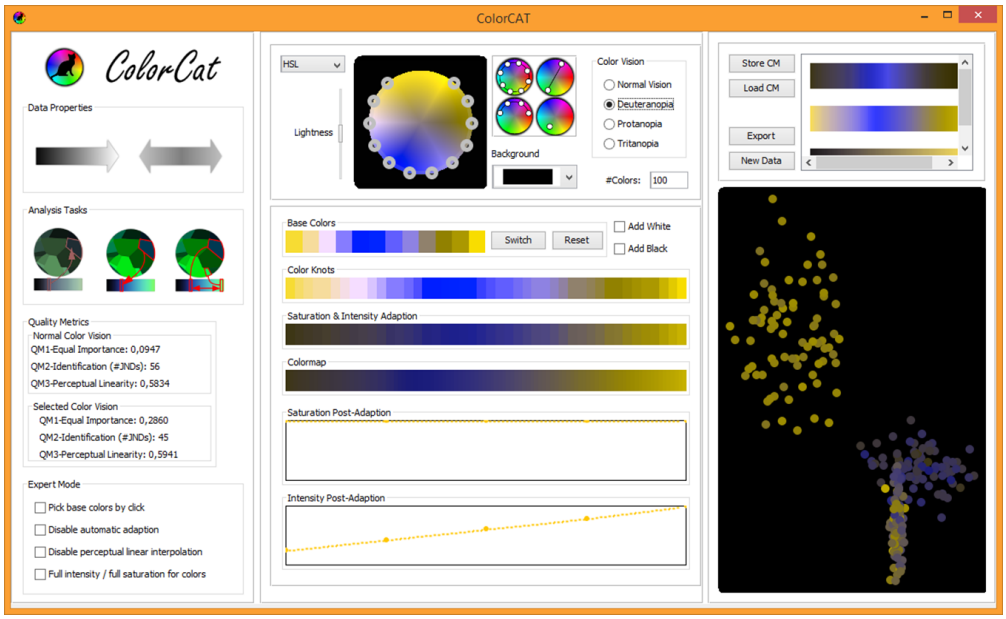


Figure 2.15: Image taken from [10]. *ColorCAT simulating red-green blindness (compare to Figure 2.12).*

Color Map Algorithm

As discussed in Section 2.3.3 (p. 57), the aim of *ColorCAT* is to integrate the visualization expert into the colormap algorithm instead of performing "black-box" optimization algorithms that find an optimal solution by minimizing the combination of quality metrics. Therefore, we provide a steerable algorithm for the visualization expert to design the colormap according to the target domain, culture, and user preferences. We exploit the heuristic guidelines discussed in Section 2.3.3 (p. 57) with the following algorithm.

The perceptual uniform color space CIECAM02-UCS (Luo et al., 2006) is used to estimate perceived color differences and thus, for creating perceptual linear colormaps. However, perceptual uniform color spaces share the problem that interpolation especially along the lightness channel often leads to colors that are undefined in RGB (Lee et al., 2012). The HSI color space (Keim, 2000) is not perceptually uniform but is defined (in RGB) for the creation of col-

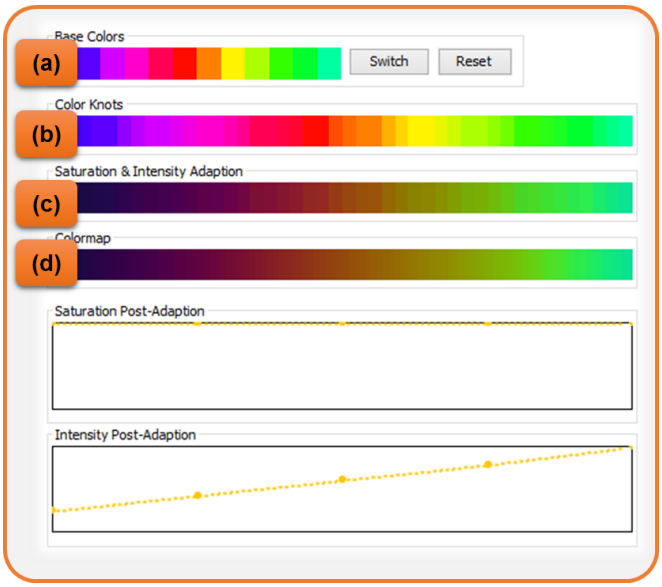


Figure 2.16: *Colormap algorithm of ColorCAT. (a) Base colors are selected by the user. (b) Base colors are extended to color knots. (c) Adapting the saturation and intensity of color knots to match the requirements of the analysis task. (d) Interpolate all adapted knots to provide a perceptual linear colormap.*

ormaps with perceptual linear increasing intensity. The trade-off is that all interpolations between colors are calculated in CIECAM02 (in order to provide perceptual linearity) and HSI is used for modeling the intensity (and saturation) properties of a colormap.

The algorithm performs the following steps (see Figure 2.16): 1) The base colors are extended by interpolation between the base colors in the HSI color space. This increases the number of "color knots" for interpolating intensity and saturation. 2) The algorithm assigns intensity and saturation to the "color knots" according to the analysis task combination and to the guidelines in Section 2.3.3 (p. 57). 3) For perceptual linearity, the algorithm computes the distances between all color knots with CIECAM02-UCS and places the knots according to their distances in the final (ordered) set of colors for the colormap. 4) The empty spaces between the knots are interpolated in CIECAM02 to provide a perceptual linear colormap.

Interactive Refinement & Interfaces

ColorCAT visualizes each step of the algorithm in order to enable the user to understand how the base colors are utilized and modified, which can be interactively changed in the color picker tool (Figure 2.14). Further, we provide an interactive spline chart that visualizes the intensity and saturation of the colormap (Figure 2.17). The user can modify the splines by interactively adding, removing, and moving control points of the intensity and saturation splines. Thus, the user can specify the data importance (see Section 2.3.1 (p. 44)) for *localization* tasks. The quality metrics panel reveals the quality of the colormap by the metrics $QM1 - QM3$ (Figure 2.12). *ColorCAT* enables the user to store alternative colormaps in a list and provides a scatterplot of continuous data, which is encoded with the selected or currently modified colormap for visual inspection.

Figure 2.17: *The user can specify the importance of data values with interactive saturation and intensity splines and highlight all values by interactive exploration, e.g., from (a) to (b).*

The background color has high impact on the foreground color perception. Colors may blend with the background or strong contrasts change the appearance of single colors as discussed in Section 2.3.3 (p. 57). Therefore, the designer can switch the background color of the scatterplot for visual inspection.

Our colormaps work best on black backgrounds. We omitted the usage of white for the colormaps, which is often used to encode extreme values but would blend with white backgrounds. *ColorCAT* exports the colormaps in different formats (RGB and CIELAB color pallets, Java and Javascript arrays) in data files, but can also export colormaps directly as JAVA classes. Exported classes can be directly used in JAVA based systems to visualize the colormap. Furthermore, the JAVA classes provide the interactive spline chart to modify the intensity and saturation properties of colormaps within the application it is applied to.

2.3.5 Case Studies

In the following, we will present two case studies to demonstrate the application of *ColorCAT*.

Visual Adverse Drug Event Detection

Adverse reactions to drugs are a major public health care issue. Currently, the Food and Drug Administration (FDA) publishes quarterly reports that typically contain on the order of 200,000 adverse incidents. In such numerous incidents, low frequency events that might be clinically highly significant, however, often remain undetected. In [9], we introduce a visual analytics system to solve this problem using (1) high scalable interfaces for analyzing correlations between a number of complex variables (e.g., drug and reaction); (2) enhanced statistical computations and interactive relevance filters to quickly identify significant events including those with a low frequency; and (3) a tight integration of expert knowledge for detecting and validating adverse drug events. (Parts of this section appeared in [9]. For the division of responsibilities and work in this publication, please refer to Section 1.5 (p. 18).)

Core of this visual analysis design is the color encoding, which communicates

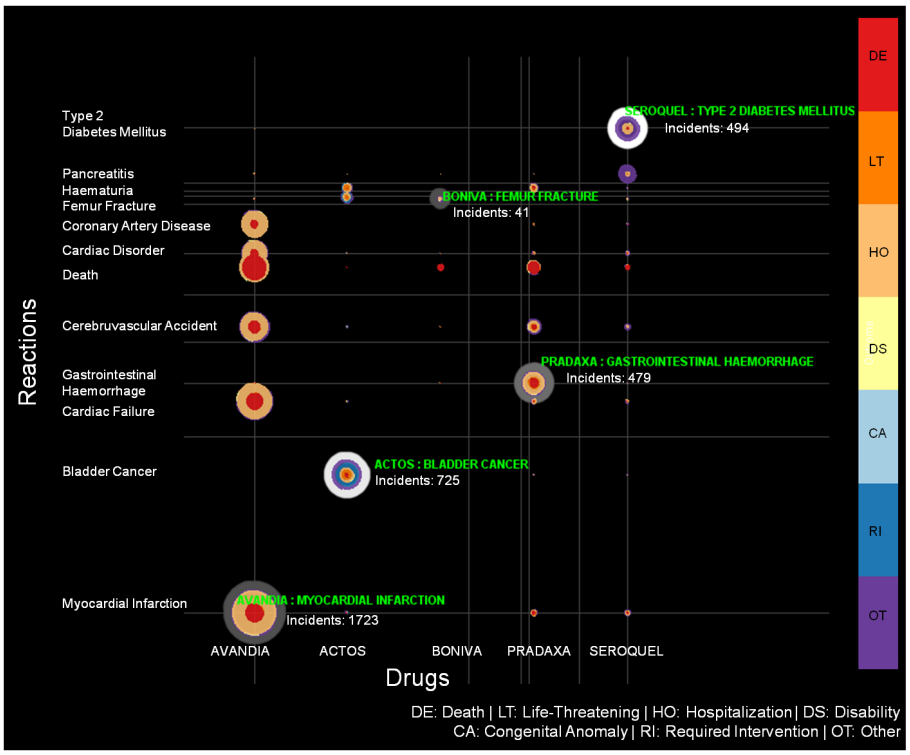


Figure 2.18: Image taken from [9]. *A non-overlapping x-y pixel interface for detecting adverse drug events (x-axis: drug name, y-axis: adverse reaction, color: outcome, white: signal strength). Each pixel represents one adverse drug-reaction event, whose color encodes the outcome in the AERS database in 2011 (3rd quarter). The visualization shows a number of known and some previously unknown adverse drug events: The known association of the drug Avandia with myocardial infarction (Nissen and Wolski, 2007) occurred 1,723 times with 280 associated deaths. The association of Actos with bladder cancer shows a high signal (white circle), which was described in the ISMP report in 2012 (ISMP QuarterWatch, 2012b). Boniva has a low frequency association with femur fracture (41 adverse events) and 42 associated deaths (ISMP QuarterWatch, 2011), an adverse drug event which has been detected by our technique. Also, it has been recently discovered that Pradaxa causes gastrointestinal hemorrhages resulting in death and hospitalization (ISMP QuarterWatch, 2012a).*

the automatic analysis results to the domain expert in high scalable overviews. We would like to refer the reader to [9] for the automatic methods for detecting potential candidates as well as for the visual analytics system, since these contributions are beyond the scope of this thesis.

Visual Encoding

After the information of "which drug caused which reaction", which is visualized in a categorical and overlapping free scatterplot of Keim et al. (2010) (Figure 2.18), the "outcome" is the most important dimension to reveal the severity of an adverse drug event. It scales from common medical investigation to the death of patients. The data type of this variable can be interpreted rather ordinal as categorical since the outcomes can be ordered according to the severity of the event.

We see different domain and application specific constraints for an appropriate color mapping. 1) The task of the analyst is the ② *identification* ($QM2$) of the outcome of an adverse drug event (ADE). 2) The data is ordered from low to high severity of ADEs and this order should be perceived by the expert (③ *comparison* task, $QM3$), which can be effectively encoded by saturation or lightness. This would highlight saturated or bright ADEs in the visualization and attract the analyst's attention towards more severe events. 3) However, steering of attention can be harmful in our particular task, since the system must treat each ADE equal in the visualization, because the impact of real significant but unexpected ADEs can only be estimated by the expert and not by the system. Therefore, highlighting of not significant ADEs should be avoided ($QM2$), which limits the exploitation of the saturation and especially of the lightness channel. 4) The ordering of events is complex, while the first three event types ("other", "required intervention", and "congenital anomaly") do not indicate that the patient's health and life was in danger, the other four types ("disability", "hospitalization", "life-threatening", and "death") clearly indicate severe outcomes. This *diverging* relation should also be encoded.

Therefore, this application requires a *semantic consistent* diverging color encoding for the metric exploration task (combination of identification and comparison). However, the variations of intensity should be avoided, which limits to express the ordering.

Varying over hues has the disadvantage that the natural ordering of hues varies widely from subject to subject. Semantics and the use of metaphors,

however, can ensure correct ordering. Our choice is the temperature metaphor; thus, we vary from cold colors (violet, blue, cyan) to warm colors (yellow, orange, red). This is also in-line with the western culture that associates severe events with red. Studies have shown that humans are more attracted to warm colors than they are to cold colors, which suppresses cold colors that are spatially close to warm colors (Wang et al., 2008). Hence, we must decrease the effect by ordering the records in the circular layouting algorithm of the generalized scatterplot (Keim et al., 2010) by decreasing severity from inner to outer radius. Thus, the inner (severe) records are more perceptually striking, but do not cover larger parts of the visualization, which reduces the bias for the analyst.

We configure *ColorCAT* to provide colors for the *identification* task and select a set of three cold colors in the hue range from $180°$ to $270°$ and four warm colors from $0°$ to $60°$ while ensuring that the colormap can also be read by color blind persons (Figure 2.19). Then, we adjust the saturation curve to create the diverging property of the colormap. We configure the algorithm to produce 12 colors in order to select the final set of colors that are visible in Figure 2.18.

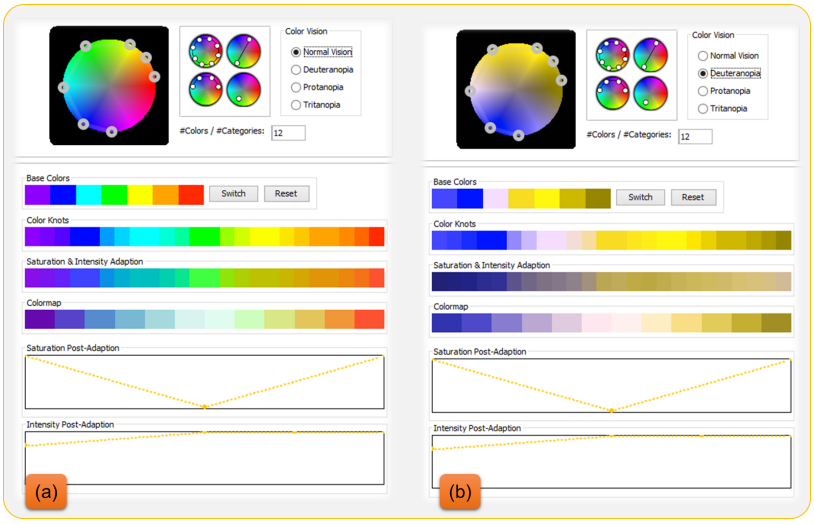


Figure 2.19: *Designing an appropriate colormap for visual adverse drug event detection with ColorCAT. (a) Normal color vision. (b) The colors can also be read by color blind persons.*

Highlighting

To show the statistical significance of each adverse drug event, we use highlighting circles in gray scales. This has two reasons: First, to encode the level of significance of the adverse event with different shades of gray. Second, the outer ring increase the size and thus, the visual importance ($QM1$) of the highlighted events in the visualization leading to an enhanced visibility of low frequency events without distorting their color. High statistical significance is encoded with white leading to the highest effect of enhancement. We are aware of the potential bias that the outer ring can have on the other categorical "outcome" colors. However, since we are only using seven colors that can be clearly separated, we consider the risk as acceptable. For further highlighting, we place labels next to the significant events.

Visual Analysis of Critical Infrastructures

The energy networks, and in particular the electricity grids, are central to the objective of ensuring sustainability, competitiveness and security of energy supply in a future transnational energy-efficient and low-carbon economy. Vulnerability in electric power systems may significantly affect security of energy supply with potential cascading impacts on other fundamental societal functions such as life/health, environment, and economy. While there exist detailed models for individual types of infrastructures such as electric power grids and mobile grids, these do not encompass the various interconnections and interdependencies to other networks. In [12, 16] we, therefore, propose a visual analytics system that provides a mechanism for conjoining the available information of different infrastructures and social media as well as mobile in-situ analysis. With the integration of these components, the system provides unified views and analytical tools for monitoring, planning, and decision support. The color encoding plays an important role in our system since it is the central mechanism that conjoins the information of different infrastructures by visualizing the status of infrastructure elements consistently. The next paragraphs focus on the color en-

coding. Since the whole system is beyond the scope of this thesis, we would like to refer the reader to [12, 16] for details. (Parts of this section appeared in [12, 16]. For the division of responsibilities and work in these publications, please refer to Section 1.5 (p. 18).)

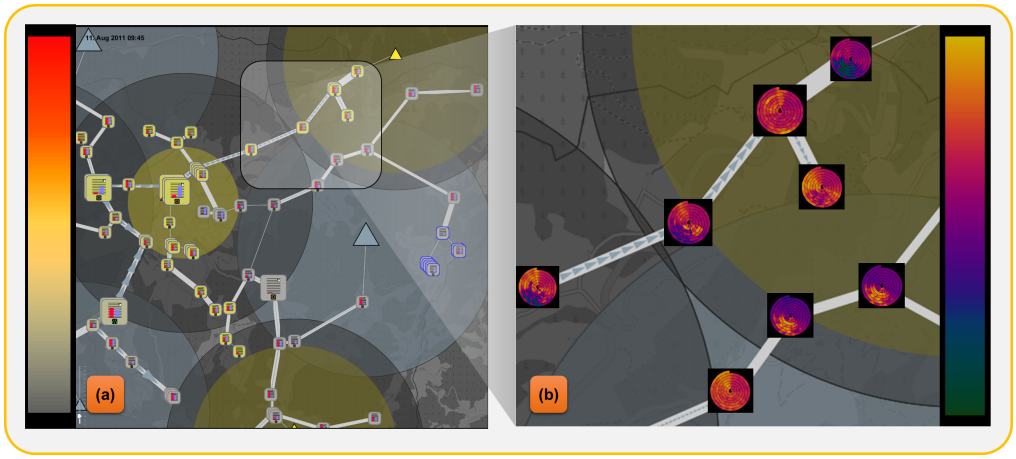


Figure 2.20: *Visual Analysis of Critical Infrastructures. (a) Monitoring the status of infrastructure elements. The colors communicate the "request for action" and severity of the event. (b) Analysis of (past) power consumption data visualized with time series spirals (Weber et al., 2001). (Note that, due to privacy reasons the images do visualize real data but are obfuscated by switching geo-locations and by aggregation.)*

Monitoring

A smart grid (energy network) typically consists of power lines at different voltage levels connected by transformer stations. These stations distribute the power over regions and supply streets, households, and industrial facilities. In our scenario, a mobile communication grid transfers information and control commands from the central control room to the electrical grid. The mobile transmitters themselves are power-supplied by common transformer stations and thus, the infrastructures are tightly interconnected.

The complex and vast amount of information of each infrastructure is reduced to the essentials, in order to enable the decision maker to understand the full crisis in its context and to detect potential cascading effects. Every infrastructure is abstracted to an undirected graph. Its nodes are represented by symbols, such

as rectangles for transformer stations and triangles for mobile transmission stations. The graph edges represent the domain dependent connections between infrastructure elements, such as power lines and mobile communication connections (see Figure 2.20(a)).

The status of each element is estimated by a state-evaluator model that is defined for each infrastructure. These models concern the actual information of the field, such as utilization and durability but also the "expected" measurements that are predicted with a simulator based on past data (see e.g., Figure 2.20(b)). A set of rules map the input of the infrastructures and anomaly detector to the states: normal (no action required), warning (action required), and alarm (immediate action required) (see Figure 2.21(a)).

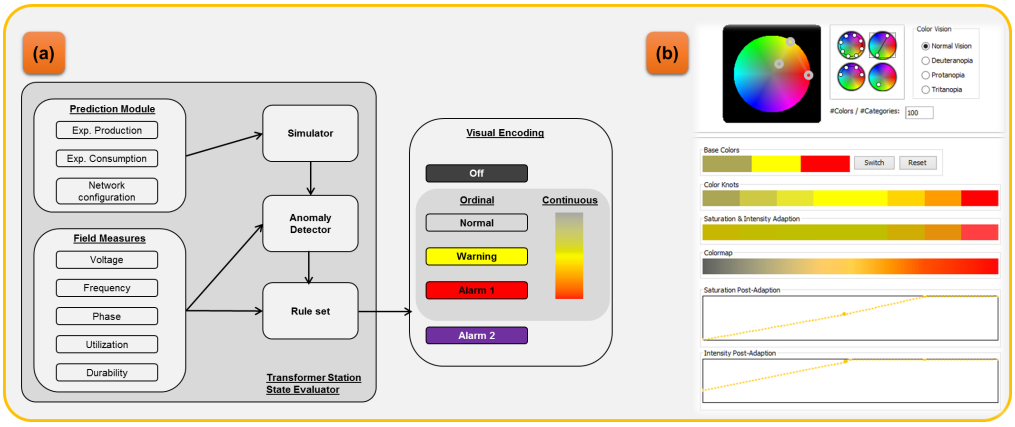


Figure 2.21: Image (a) taken from [16]. *(a) Station state evaluators consider the incoming measurements and the comparison to the expected behavior, which reveals anomalies. A set of rules maps this input to color that expresses the status of the element. (b) ColorCAT is used to design a colormap that reveals the "request of action" and the severity of events.*

In this monitoring application, the analyst aims to ①*localize* severe events on the display. Severe events must be perceptually highlighted on the dark background whereas less important events should have less visual impact. Thus, the attention of the analyst is steered towards events for which actions are required. Further, the analyst has to ②*identify* if immediate action is required or not, such that he/she can ③*compare* events to perform actions on the most critical events first. Thus, the coloring must satisfy the requirements of all ele-

mentary analysis tasks. We select yellow and red to communicate the different categories of "required action" for *semantic consistency*. The colors increase in intensity and saturation such that severe events are perceptually highlighted for *localization* on the dark background whereas less important events do have less visual impact and blend with the background. The maximum saturation for yellow and red lets the analyst effectively identify the "request for action" and the perceptual linearity allows the analyst to *compare* the states of elements.

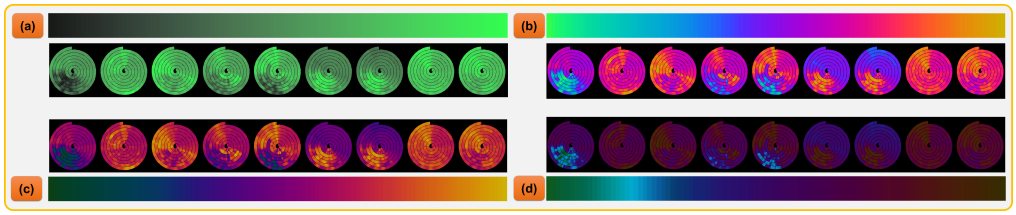


Figure 2.22: *Effective colormaps for different tasks in analyzing power consumption data. (a) ③Elementary comparison. (b) ②Elementary identification. (c) ㉓Metric exploration. (d) ⑫Elementary exploration.*

Analyzing Power Consumption Data

Another important analysis task in this application is the analysis of power consumption measurements over time. Therefore, the overview can be switched to visualize the power consumption measurements of the last seven days to reveal consumption patterns in the overview (see Figure 2.20(b)). The spiral visualization provides effective overviews of seasonal time series data especially for revealing periodic patterns. Each $360°$ encode one day of past power consumption data. Figure 2.22 reveals the strength and weaknesses of colormaps for the different tasks. With (a), the analyst can spot the trends by *pre-attentively* comparing data points. (b) shows far more details that can be identified, however, may bias the user in comparing data values. (c) combines identification and comparison for the user to explore the metrics of the time series data. (d) allows localizing and identifying data values interactively. The user can interactively adapt the colormap that best fits the current analysis task.

2.4 Color Encoding for (High-Dimensional) Data Relations

An important example where expressing additional visual dimensions is direly needed is the analysis of high-dimensional data. The property of perceptual uniformity is desirable in these applications, because this enables the user to *pre-attentively* perceive clusters and relationships among multidimensional data points.

Multi-dimensional encodings with color were evaluated by Ware and Beatty (1988) and Wainer and Francolini (1980). Wainer and Francolini (1980) evaluated two-dimensional encodings and found that 2D colormaps are unintelligible to read metrics of both dimensions. Ware and Beatty (1988) performed an experiment, in which five dimensional data was mapped to two spatial and three color dimensions. The results indicated that each additional color dimension is as useful as an additional spatial dimension for cluster identification since data relations are preserved. This controversy highlights that *effectiveness* depends on the analysis task. In high-dimensional data analysis, the focus is typically on exploring the relations of data items and not on encoding data dimensions. (The next section describes requirements for synoptic analysis tasks in detail.)

Perceptual uniformity is already modeled in color spaces such as CIELAB (see Section 1.6.1 (p. 24)). If the distances in the data space are mapped to perceptual distances in the color space, the analysts will perceive the relations of data items *pre-attentively* by interpreting the perceptual similarity of their colors. In this case, the color mapping is not bound to a fixed number of dimensions and is able to encode high-dimensional data relations. Unfortunately, only a subspace of CIELAB can be visualized on current displays. This subspace (or

bounds) is of non-rectangular shape that makes interpolation and other arithmetics for color mapping very complex (see Figure 2.23C). Rectangular parts of this subspace as defined by a maximum surrounded box provide perceptual linear mappings but result in fewer discriminable colors (see Figure 2.23B).

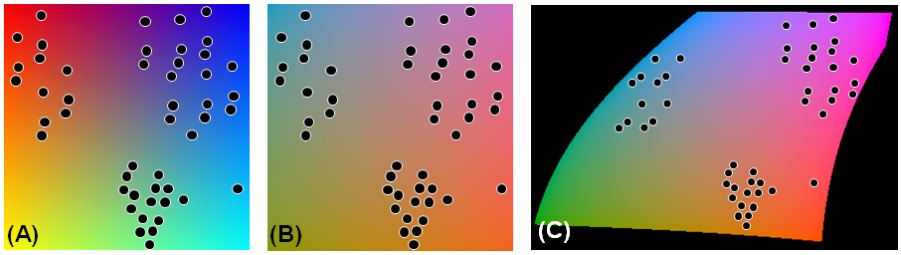


Figure 2.23: Image taken from [8]. *Two dimensional colormaps. The position of black dots represent the color of multidimensional data points.* ***(A)*** *2D colormap in RGB: colors are saturated, however, not perceptually uniform;* ***(B)*** *Rectangular sub plane of CIELAB: perceptually uniform, but less saturated colors;* ***(C)*** *Our method: saturated and perceptually uniform colors.*

The current state-of-the-art is to project high dimensional data to a lower (two) dimensional space and then scale the projected data to fit a two dimensional color map (Figure 2.24). Most methods interpolate in RGB between fixed color anchors in the corners (Himberg, 2000) (Figure 2.23A). This results in highly discriminable colors but these color maps are not perceptually linear. Furthermore, the user may group data points of the same cluster differently, which may result in a misinterpretation of clusters (e.g., in Figure 2.23A clusters span over two or more color hues). Therefore, these approaches do not sufficiently support visual analysis tasks. For example, identification of clusters requires that each cluster is perceptually distinct to other clusters but also that data relations between and within clusters are faithfully represented.

Kaski et al. (2000) previously introduced a method that projects high dimensional data with self-organizing maps to two dimensions and fits the data into the bounds of CIELAB (see Figure 2.25). The color assignment supports the user in recognizing clusters and preserves the relationships of clusters while maximizing the exploitation of the color space. The result of this algorithm is dominated by the clustering and projection of the self-organizing map, which

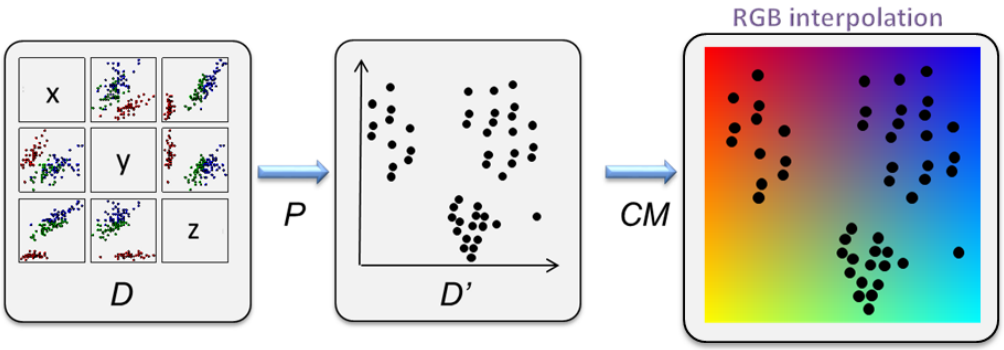


Figure 2.24: *High-dimensional data is projected into two dimensions and then scaled to a two-dimensional colormap.*

may mask high frequencies or arbitrary clusters in the data. Therefore, it may fail in synoptic analysis tasks such as cluster identification, in which clusters are unknown and should be perceptually identified by the user, or synoptic localization, in which clusters and classes should be localized on the display but may have been masked by the self-organizing map.

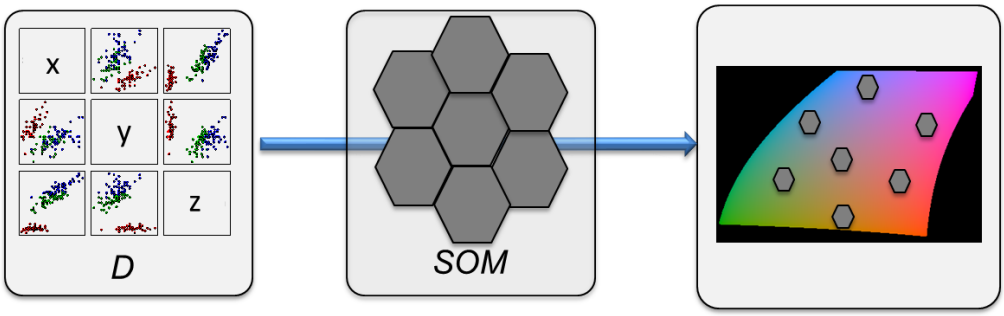


Figure 2.25: *Method of Kaski et al. (2000). High dimensional data is projected by a self-organizing map into the CIELAB color space. All data items of one cluster share the same color.*

In this section, we revisit the method of Kaski et al. (2000) and adjust it to the requirements of visual analysis tasks (Figure 2.26). Our method provides improved color mapping for high dimensional data points, which can be used in any visual design since color as an additional design variable that is most effective in combination with other visual variables such as position. A result of our method is illustrated in Figure 2.27. We show in a user study that this method

outperforms the state-of-the-art but highlight also further research questions.

We claim the following contributions: 1) **generalization** of the method with further projection methods, extension to 3D target color spaces, and guidelines for analysis tasks; 2) **efficient heuristics** for practical use; 3) **cost functions** to further support synoptic analysis tasks; 4) **evaluation** of different configurations and methods in a user study.

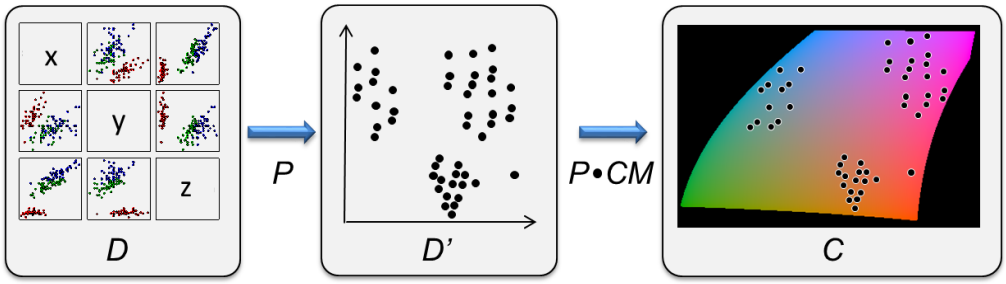


Figure 2.26: Image taken from [8]. *Our generalization of the method of Kaski et al. (2000). Projection methods P map high-dimensional data objects of data space D to perceptual uniform color spaces C with regards to specific analysis tasks such as synoptic identification of clusters or localization of classes.*

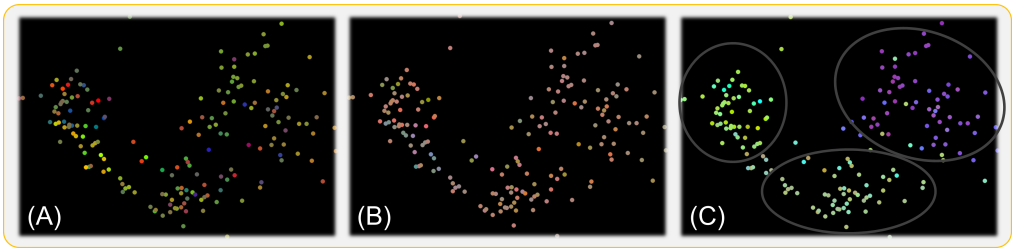


Figure 2.27: Image taken from [8]. *Wine data set (Bache and Lichman, 2013). 13 attributes describe three classes of wines. The data is projected with MDS to four dimensions and visualized in the scatter plots (x-, y-axis and two dimensions are mapped to color).* ***(A)*** *2D RGB color map: classes are not separated and colors reveal (wrong) large distance between data points;* ***(B)*** *CIELAB sub plane: distances are preserved but classes are not separated;* ***(C)*** *Our Method: Three classes are separated and local distances of class elements are preserved.*

2.4.1 Requirements for Synoptic Analysis Tasks

These tasks are focused on the analysis of sets or groups of objects e.g., identifying clusters in the visualization according to their color-encoded properties. However, the visualization of groups still represents single data elements. Thus, requirements that support elementary analysis tasks (ER1-ER3, see Section 2.3.1 (p. 44)) also serve as a basis for synoptic tasks. The focus is on the relation of data objects, which implies that the task of visual comparison is the base for this group of tasks.

SR1 Global **Synoptic Comparison** is the task to relate two or more sets of objects. This implies that the properties within a set or cluster are preserved but also the relations between clusters. This transfers **ER3** (elementary comparison) to two different levels. The color mapping should preserve global (between sets) and local (within sets) data properties. However, if global relations are in the focus, the requirement of preserving *all* pairwise distances (**ER3**) can be relaxed since local data-properties of single elements are not compared across sets. Instead, the task to relate or to compare sets of objects requires that the relations between sets are preserved in the coloring.

SR1 Local **Synoptic Comparison.** Local relations within sets of elements have to be preserved if the focus is on analyzing local properties of sets (e.g., the relations of elements within one cluster). Therefore, the coloring can also relax **ER3** of preserving *all* pair-wise distances and focus on preserving local relations by discarding global (between sets) relations. Thus, **SR1 Local** can be defined to preserve the relations of an object to its neighborhood in the color mapping.

SR2 **Synoptic Localization** defines the task to detect an *a priori*-known group or class in the visualization. This implies that one group has to be visually differentiable against all other groups. For example, one class is encoded

in blue whereas the other class is encoded in red. The color mapping has to provide colors that separate (known) groups or classes of data objects by color (**ER1**, elementary localization) and still preserve the data relations within the set (**SR1 Local**).

SR3 In **Synoptic Identification**, the task is to identify (unknown) groups or classes in the visualization. For example, an analyst identifies a cluster of objects that share similar reddish colors and thus, sharing similar characteristics. Therefore, the analysts' attention should not be dragged towards other distracting visual encodings of certain groups, which implies that the color encoding should assign colors with equal visual importance (**ER2**, elementary identification). Further, the visual encoding has to preserve all pair-wise data relations (**ER3**) in order to enable the perception of data clusters as visual clusters.

2.4.2 Quality Metrics for Effective Color Encoding of Data Relations

---

For measuring the effectiveness of color encodings for synoptic analysis tasks, we propose novel quality metrics that are based on the following definition.

Definitions (see Figure 2.26 (p. 79) as illustration): D is the set of all model vectors $m_i \in \mathbb{R}^m$ describing all data elements i. C is the set of all colors $c_i \in \mathbb{R}^n$ in the target color space. $P : \mathbb{R}^m \mapsto \mathbb{R}^n$ is the projection of the high-dimensional model vectors in the lower dimensional target space D'. D' being the set of model vectors $m'_i \in \mathbb{R}^n$ (note that $D' \neq C$). $CM : \mathbb{R}^n \mapsto C$ is the color assignment of m'_i to color c_i. G is the set of clusters in D with $g(i)$ being the cluster of data element i. H_x being the convex hull and $V(H_x)$ the volume of a set or cluster x (e.g., $V(H_D)$ represents the volume of the convex hull of D in $\mathbb{R}^m$).

Faithful Preservation of Data Relations.

For synoptic comparison (**SR1**), it is required that data distances match the perceptual distances. The according quality measure for a projection from high- to low-dimensional space is, therefore, the preservation of all relative distances (**SR1 global**) that can, for example, be measured by the Sammon's stress measure (Sammon, 1969) with Eq. 2.18. However, the preservation of all pairwise distances is typically impossible in large data sets. Therefore, Kaski et al. (2000) preserve the relative distances within a cluster (**SR1 local**) to increase the accuracy of the projection locally (Eq. 2.19).

$$f_1 = \sum_{i \in D} \sum_{j \neq i} \frac{\left(d\left(m_i, m_j\right) - d\left(m_i', m_j'\right)\right)^2}{d\left(m_i, m_j\right)} \tag{2.18}$$

$$f_2 = \sum_{i \in D} \sum_{j \in g(i)} \left(d\left(m_i, m_j\right) - d\left(m_i', m_j'\right)\right)^2 \tag{2.19}$$

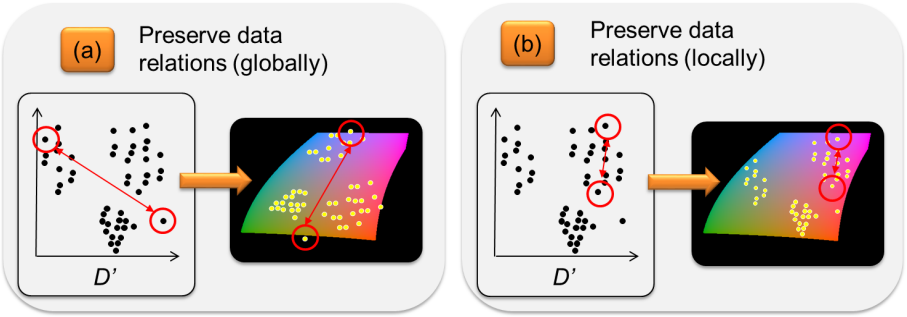


Figure 2.28: *Preservation of data relations for synoptic comparison (based on the assumption that the input data is faithfully projected from high dimensional spaces to the low dimensional space D'). **(a)** exemplifies that all pairwise relations can be preserved globally (**SR1 global**) to minimize f_1. **(b)** focuses on the preservation of local relations to minimize f_2 (**SR1 local**, relations between clusters may be skewed but relations within clusters are preserved).*

Expressive Color Space Exploitation.

Another important property of effective color encodings is that the mapping exploits as much of the color space as possible in order to provide distinguish-

able colors for synoptic identification (**SR3**). Kaski et al. (2000) rigidly scale the vectors m'_i with a parameter k. The scaling parameter k is increased to let D' occupy more of the available color space C. The original method of Kaski et al. (2000) estimates the distance of m'_i to its perceptually closest color c_i that can be displayed on the output device. This does not measure the exploitation of the color space. It measures the distortion of CIELAB colors that lay beyond the color space bounds (Eq. 2.20). The exploitation of the color space can be measured by the overlap of the color space in $\mathbb{R}^n$ and the projected data $D' \in \mathbb{R}^n$. This can be approximated by computing the volume of the intersection of the convex hulls of $H_{D'}$ and H_C (2.21) (see Section 2.4.3 p. 88 for algorithms that estimate convex hulls and overlaps of convex hulls).

$$f_3 = \sum_{i \in D} d\left(m'_i, c_i\right) \tag{2.20}$$

$$f_4 \approx 1 \,/\, V\left(H_C \cap H_{D'}\right) \tag{2.21}$$

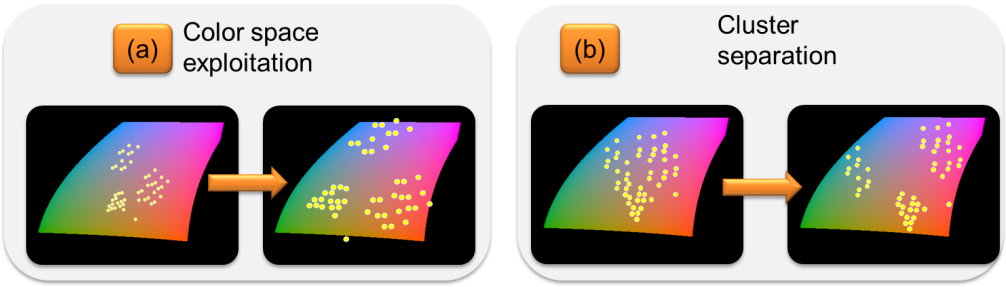


Figure 2.29: ***(a)*** *Maximizing the exploitation of the color space is important for synoptic identification.* ***(b)*** *Separation of known clusters and classes is very effective for synoptic localization.*

Preservation of clusters.

Preserving the local distances within a known cluster and ignoring the interrelations of clusters makes the color mapping very flexible. The data can adapt to the non-linear shape of the color space, which separates clusters well for synoptic localization (**SR2**). However, if the task requires also to perceive interrelations of clusters (**SR3**), this method will produce misleading results. Kaski

et al. (2000) introduced a heuristic that measures the "orderliness" of clusters based on a SOM grid. We propose a different function that preserves the relative distances of cluster centroids $\bar{m_r}$ with $r \in G$ (Eq. 2.22), because the heuristic of Kaski et al. (2000) cannot be applied in high-dimensional spaces.

$$f_5 = \sum_{\bar{m_r}, r \in G} \sum_{\bar{m_s} \neq \bar{m_r}} \left(d\left(\bar{m_r}, \bar{m_s}\right) - d\left(\bar{m_r}', \bar{m_s}'\right) \right)^2 \tag{2.22}$$

Further, the original method of Kaski et al. (2000) does not measure how well clusters are separated or if they overlap, which is important for synoptic localization (**SR2**). The overlap can be approximated with the inverse centroid distance (Eq. 2.23) or the intersection of convex hulls (Eq. 2.24). Another issue in visualizing clusters with color is that we will overestimate the number of clusters or see noise if there are only few present (Ware and Beatty, 1988). Our cognition tries to differentiate between groups and objects based on their color (hue). If a cluster spans over the whole color space, it is likely that it is perceived as multiple clusters. With cost function f_8 (Eq. 2.25) we measure, therefore, the pairwise color distances of cluster elements that are higher than a threshold t. We informally determined a value of $t = 30$ in CIELAB as an appropriate value to perceive clusters correctly. Thereby, we systematically decreased t and checked the results on different sample cluster images. The value of $t = 30$ was appropriate in our experience, however, a formal evaluation remains future work.

$$f_6 \approx 1 \,/ \sum_{\bar{m_r}, r \in G} \sum_{\bar{m_s} \neq \bar{m_r}} d\left(\bar{m_r}', \bar{m_s}'\right) \tag{2.23}$$

$$f_7 \approx \sum_{r \in G} \sum_{s \neq r} V\left(H_r \cap H_s\right) \tag{2.24}$$

$$f_8 = \sum_{r \in G} \sum_{i,j \in r} \sum_{j \neq i} max\left(d\left(m_i', m_j'\right) - t, 0\right) \tag{2.25}$$

2.4.3 Optimization of Effective Color Encodings for Data Relations

The optimization goal is to minimize the multi-objective cost functions. We, therefore, scalarize and sum the functions (2.26). Note, that this may be different with other optimization methods. Scalar α_i is used to make the cost functions comparable. This parameter can be estimated, for example, by evaluating several random solutions and normalizing all cost functions according to the median costs m_i of the random solutions ($\alpha_i = m_i^{-1}$). λ_i steers the influence of the cost function i on the mapping and configures the method for different analysis tasks. Details can be found in Section 2.4.4.

$$f = \sum_{i=1}^{8} \lambda_i \cdot \alpha_i \cdot f_i \tag{2.26}$$

The optimization goal can be reached by minimizing the sum of cost functions by a variety of optimization algorithms. Kaski et al. (2000) use a stochastic gradient method. We found that *particle swarming* (Kennedy et al., 1995) provided good results. However, we consider the choice of the optimization algorithm as interchangeable part of our method.

The optimization goal $min(f)$ has several issues: 1) f is not continuous so that f' can only be approximated; 2) in high dimensional spaces f_1 and f_2 suffer under the curse of high dimensionality. Sophisticated projections P exist that effectively map $\mathbb{R}^m$ to $\mathbb{R}^3$. We, in practice, use a standard projection technique P such as MDS of Brandes and Pich (2007) and project the data to $\mathbb{R}^3$. The fitting to CIELAB is then applied in a post-processing step (see Figure 2.26 (p. 79)). Global and/or local distances can be preserved by P. Therefore, a heuristic can use translation (in three dimensions), scaling (one parameter for all dimensions), and rotation (about three axis; centers as fix points) on the projected data D' or on clusters in D' to minimize the cost functions. Thereby, each data vector m'_i in data space D' is mapped by affine transfor-

mation to the color space with the parameters t_x, t_y, t_z (for translation), s (for scaling), and $\alpha_x, \alpha_y, \alpha_z$ (for rotation) in Eq. 2.27. This results in only seven parameters that have to be estimated by the optimizer to map the whole data D' into the color space (if all pairwise distances shall be preserved) or seven parameters per set (if the task is focused on the localization of classes). In *particle swarming*, the parameters are found by initializing the position (parameter vector) of each particle (randomly) with random velocities. The position of each particle is updated in each iteration with its velocity influenced by the own best known position and the global best known position. Once all particles converge to a position, the optimal parameters (best position) are found. Details are described by Kennedy et al. (1995).

$$\vec{c_i} = T(\vec{t}) \cdot S(s) \cdot R_x(\alpha_x) \cdot R_y(\alpha_y) \cdot R_z(\alpha_z) \cdot \vec{m_i'} \tag{2.27}$$

$$\vec{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \\ 1 \end{pmatrix} \tag{2.28}$$

$$T(\vec{t}) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.29}$$

$$S(s) = \begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.30}$$

$$R_x(\alpha_x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha_x) & -sin(\alpha_x) & 0 \\ 0 & sin(\alpha_x) & cos(\alpha_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.31}$$

$$R_y(\alpha_y) = \begin{pmatrix} cos(\alpha_y) & 0 & sin(\alpha_y) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\alpha_y) & 0 & cos(\alpha_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.32}$$

$$R_z(\alpha_z) = \begin{pmatrix} cos(\alpha_z) & -sin(\alpha_z) & 0 & 0 \\ sin(\alpha_z) & cos(\alpha_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.33}$$

Configuration for Analysis Tasks.

Our guidelines are summarized and illustrated in Table 2.30. f_3 and f_4 are independent of the task and should always be activated. However, if the task also implies the comparison of single data items or *synoptic identification*, $f1$ will ensure that perceptual relations are equal to data relations and that perceptually accurate mappings are provided.

When clusters or classes are known *a priori* and should be perceptually preserved for *synoptic localization*, f_6, f_7, and f_8 should be activated since they support localization of clusters. f_1 should be deactivated since it tries to preserve *all* pair-wise distances. To preserve the data relations within clusters or classes f_1 should be replaced by f_2. If clusters or classes need to be compared (*synoptic comparison*), f_5 should be activated as well since it ensures that the relations between clusters are preserved.

Task / Costfunction	Elementary		Synoptic		
	SR1 Compare data points globally	SR1 Compare data points locally (cluster)	SR3 Identify clusters	SR1 Lookup classes	SR1 Compare clusters
f1	✔	✘	✔	✘	✘
f2	✘	✔	✘	(✔)	(✔)
f5	✘	(✔)	✘	(✔)	✔
f6	✘	(✔)	✘	✔	✔
f7	✘	(✔)	✘	✔	✔
f8	✘	(✔)	✘	✔	✔

✔ Activated ✘ Deactivated (✔) Optional

Figure 2.30: Image taken from [8]. *Combination of cost function for visual analysis tasks.*

Heuristics for Initialization

In any case, a good initialization for optimization can be provided by affine transformations of the projected data D' to C by matching the centers, scales and rotations of D' to C [1]. The initialization can further be refined by stepwise increasing/decreasing the scale of D' in all three dimensions until f_3 and f_4 are minimized. This heuristic already provides good results for the preservation of distances and identification of clusters. We use this as a starting point for *particle swarming* to optimize the parameter vector and project D' to C by minimizing the cost functions.

If the focus is on cluster separation we can use the same initialization mentioned before, however, scale D' in all dimensions to a minimum. Then expand/scale the clusters and move each cluster away from the others. The cluster centroid m_i^{t+1} (in iteration $t+1$) can be estimated with Eq.2.34, which assigns to far cluster higher weights.

$$m_i^{t+1} = m_i^t + \sum_{j \neq i} \frac{d(m_i^t, m_j^t)}{\sum_{j \neq i} d(m_i^t, m_j^t)} (m_i^t - m_j^t) \tag{2.34}$$

Volume and Intersection of Convex Hulls

Many of the cost functions are based on estimating the volume and intersection of convex hulls in the three dimensional color space. Performing the optimization and transformations on the convex hulls of the data set, clusters, and the color space makes this algorithm scalable for large data sets. Quickhull (Barber et al., 1996) is an efficient algorithm that estimates the convex hull in $\mathcal{O}(n \cdot log(n))$ with n being the number of data points.

The CIELAB color space can be sampled in each dimension with a sampling

[1] rotation is estimated by PCA of xy, xz and yz in the three dimensional space

interval of s ($s = 5$, see below). The Quickhull algorithm can then be computed on the samples of the color space to estimate its convex hull. The data exploits the color space if the convex hulls overlap to a high degree.

Estimating the volume of convex hulls and especially estimating the overlap of convex hulls is complex. Ahn et al. (2013) present an algorithm that estimates the maximum overlap of polytopes within $\mathcal{O}(p \cdot log(p))$ with p being the number of points of the convex hull. Further, it estimates the translation realizing the maximum overlap; scaling and rotation are not supported. An efficient shortcut to these algorithms are bounding boxes, which can be estimated in $\mathcal{O}(n)$; and the overlap as well as the volume can be computed in $\mathcal{O}(1)$. This efficient approach has the disadvantage that bounding boxes are inaccurate (e.g., a sphere of points and its diagonal points share the same bounding box) but also provided satisfactory results in our experience. However, convex hulls and bounding boxes are sensitive to outliers, which may be desired for "global" outlier detection but biases other tasks because most of the color space may remain unused. The same issue is also occurring for single clusters, since outliers cause empty spaces in a convex hull.

Therefore, we present an outlier-insensitive method to estimate color space exploitation and overlap of clusters, which is simple to re-implement: We construct a k-d-tree (Bentley, 1975) with the samples of the color space in $\mathcal{O}(n \cdot log(n))$. In order to measure the color space exploitation, we estimate the nearest neighbor cs_n of the color samples for each data point i. If the color difference is smaller or equal to the just-noticeable-difference $\Delta E(i, cs_n) \leq JND$ ($JND \approx 2.3$) (Mahy et al., 1994) then i occupies cs_n (therefore, the CIELAB color space should be sampled with $s = 5 \approx 2JND$). The number of (distinct) occupied cs_n estimates the number of JNDs used for encoding the data. If the color differences between all occupied color samples is high, then the mapping exploits much of the color space. Thereby, the sum of color differences of all cs_n approximates the color space exploitation.

In order to measure the overlap of clusters, we estimate the set of occupied color samples cs_n for both clusters and then count the number of samples that are occupied by both clusters. This algorithm is simple to re-implement, is

not sensitive to outliers, enables color mappings for color blind persons (see below), and has a complexity of $\mathcal{O}(n \cdot log(n))$. Therefore, the algorithm is efficient enough for most applications, however, does not scale for large data sets, which could be overcome with sampling of the data set. For the evaluation in Section 2.4.4 (p. 91), we apply this k-d-tree approach in the method to represent high-dimensional data relations with color.

Color Blindness

Our method is designed for normal color vision since it projects data points into perceptual uniform color spaces based on normal color vision. We argue, that our method provides effective color encodings for color vision deficiencies (CVD) if a perceptual uniform color space for CVD is applied. But, according to our literature research, there exist no such color space. However, there are sophisticated models by Brettel et al. (1997) and Machado et al. (2009) that can simulate CVD.

The kd-tree approach for measuring color space exploitation can be used to approximate a perceptual uniform color space for CVD by these CVD-models. The color blind version of our method takes the samples of the perceptual uniform color spaces cs and predicts with the method of Brettel et al. (1997) how these colors appear to a color blind person cs'. Then, the kd-tree is created by these simulated samples, however, samples are discarded if the color difference to the closest sample (that is already contained in the tree) is below the just-noticeable-difference. This is valid since the models of simulating CVD were constructed by experiments that measured the similarity of colors for color blind persons. Further, this ensures that the k-d-tree approach to measure color space exploitation is valid, since the number of occupied samples is counted to measure the overlap of color space and data space. The rest of the algorithm remains similar.

Figure 2.31 shows the result of the algorithm with simulated deuteranopia and tritanopia applied on the wine data set (see also Figure 2.27 (p. 79)). The colors

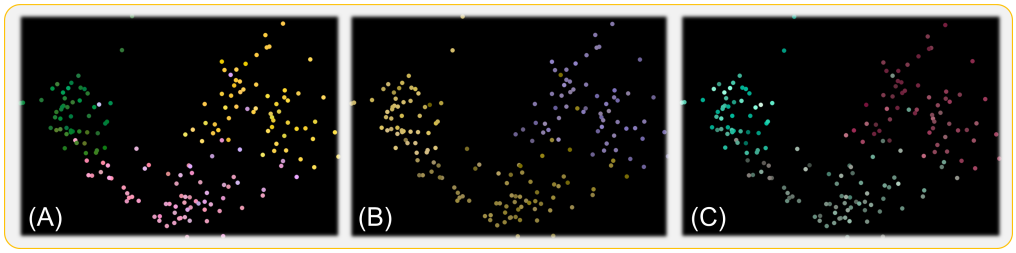


Figure 2.31: *Wine data set (Bache and Lichman, 2013). 13 attributes describe three classes of wines, which become visible with our method even for color blind persons. The data is projected with MDS to five dimensions and visualized in the scatterplots (x-, y-axis and three dimensions are mapped to color). (A) Our method for normal color vision, (B) for persons with deuteranopia (red-green blindness), and (C) for persons with tritanopia.*

in (B) and (C) appear dull to persons with normal color vision but the three classes of wine can still be estimated and the variation within the classes is also visible. Note, that this approach was only evaluated informally by repeating the experiment in the next section with two color blind persons. In this (not representative) sample, we found that this method outperforms state-of-the-art colormaps with similar results and implications as for normal color vision. A representative user study covering different types of color vision deficiencies remains future work.

2.4.4 Evaluation

Goal and Task.

We evaluated our method empirically with an experiment introduced by Ware and Beatty (1988). The goal was to measure the accuracy of users identifying the number of clusters in a visualization. The participants were shown a multi-dimensional data set in a scatter plot (as in Figure 2.27). Two spatial dimensions were encoded by x- and y-axis and two or three dimensions were encoded by

color (note, that we reduced the number of dimensions in order to be comparable with related methods). The participants were asked to estimate the number of clusters in each scatter plot (see Figure 2.32). Note, that counting the number of clusters is not trivial and involves elementary and synoptic tasks (see Table 2.30). The participant has to compare the spatial and color distribution of the data points, which is the elementary task of comparing data points globally (**SR1 global**). Further, the participant has to group the data points and to differentiate between spatial distribution and color since clusters may overlap spatially or in the color space. With this, the participant identifies clusters (**SR3**) and is able to count the number of clusters in the plot.

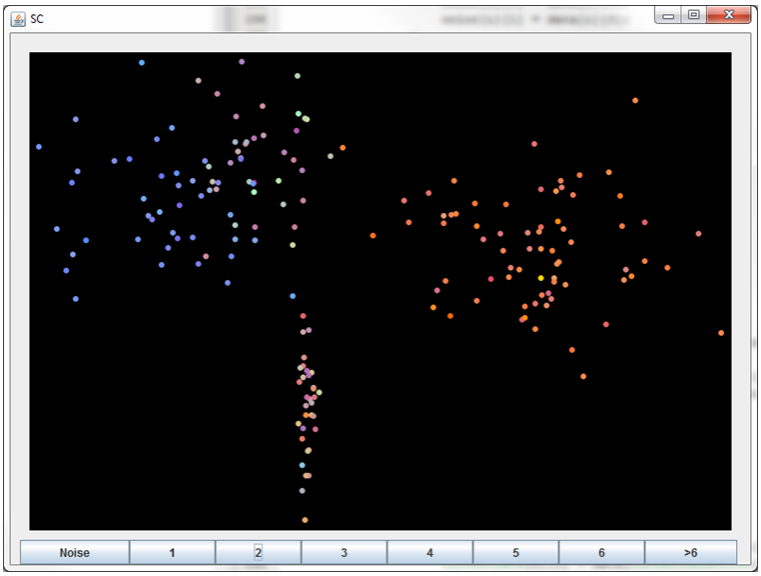


Figure 2.32: *Experiment task to identify the number of high-dimensional clusters within a color encoded scatterplot. Two spatial dimensions were encoded by x- and y-axis and the other dimensions were encoded by color.*

Experiment Factors.

We evaluated seven approaches to map high-dimensional data relations to color (three state-of-the-art techniques and four different configurations of our method).

Our method can be configured in multiple ways, however, we selected two versions. One was configured for the elementary comparison task (**SR1 global**) and synoptic identification (**SR3**). The other was configured to preserve known clusters (**SR1 local** and **SR2**, see Table 2.30 (p. 87)).

For four dimensional data, we used our method with a fixed lightness of $L = 60$ (2D version) to map dimensions 3 and 4 directly to the CIELAB color space (dimensions 1 and 2 are mapped to the axis of the scatterplot). As state-of-the-art methods for the 4-D cases, we applied two-dimensional color maps (the best of the state-of-the-art) in RGB and CIELAB (see below).

For five dimensional data we applied the method of Ware and Beatty (1988) to map dimensions 3-5 directly to red, green and blue as state-of-the-art. For the 5-D case, we project with our method dimensions 3-5 to the full CIELAB space (3D version).

State-of-the-Art 2D colormaps

Bernard *et al.* [1] survey the research field that proposes or applies two-dimensional colormaps. For their quality assessment, they propose quality metrics for static two-dimensional colormaps. The metrics are based on the definition of requirements for analysis task that are discussed in this thesis. Their metrics had to be adapted because the metrics in this thesis are data-depended and not directly applicable to static 2D colormaps that are data-independently designed. Figure 2.33 shows the results of the quality assessment.

Based on their quality metrics, a two-dimensional colormap is designed as follows: The colormap must enable intuitive and accurate readings in order to express the metrics of similarity. On the one hand, it should exploit a maximum of different colors. On the other hand, the user must be able to estimate the approximate distance between two objects correctly, which requires a perceptual uniform interpolation in CIELAB. To design a perceptual uniform 2D colormap we use four perceptual distant colors and interpolate between these colors, namely yellow, cyan, red and blue. The goal of this selection is to separate the colormap into complementary color tones and from saturated (bottom) to intense (top) colors.

Title	Color Space	Image	Black Background								White Background								
			ER1 Elementary Lookup	ER2 Equal visual importance	ER3 Elementary Comparison	SR1 Synoptic Localization	SR2 Synoptic Identification	SR3 Synoptic Comp. (local)	SR4 Synoptic Comp. (global)		ER1 Elementary Lookup	ER2 Equal visual importance	ER3 Elementary Comparison	SR1 Synoptic Localization	SR2 Synoptic Identification	SR3 Synoptic Comp. (local)	SR4 Synoptic Comp. (global)		Min (SR1, SR2) MotionExplorer[55] (UseCase)
BCP 37	sRGB		∉	++	+	∉	∉	∉	∉		∉	++	+	∉	∉	∉	∉		∉
Baum et al.	sRGB		∉	∉	∉	∉	∉	∉	∉		++	∉	∉	∉	∉	∉	∉		∉
Bremm et al. (regular)	L*a*b		+	++	++	++	++	++	Ø		+	++	++	++	++	++	Ø		++
Bremm et al. (stretched)	L*a*b		∉	++	∉	∉	∉	∉	∉		∉	++	∉	∉	∉	∉	∉		∉
Constant Blue	sRGB		−	−	−	−	−	Ø	Ø		−	−	−	−	−	−	Ø		−
Constant Green	sRGB		∉	+	Ø	∉	∉	∉	∉		∉	+	−	∉	∉	∉	∉		∉
Constant Red	sRGB		Ø	∉	−	∉	∉	∉	∉		+	∉	−	∉	∉	∉	∉		∉
Cube Diagonal Cut BCYR	sRGB		++	−	Ø	+	Ø	Ø	+		++	−	Ø	++	Ø	−	+		Ø
Cube Diagonal Cut BMYG	sRGB		+	∉	∉	∉	∉	∉	∉		+	∉	∉	∉	∉	∉	∉		∉
Four Corners R-B-G-Y	sRGB		+	∉	∉	∉	∉	∉	∉		∉	∉	∉	∉	∉	∉	∉		∉
Griscom and Palmer	L*a*b		∉	++	∉	∉	∉	∉	∉		∉	++	∉	∉	∉	∉	∉		∉
Guo et al. (Gaussian)	L*a*b		∉	+	∉	∉	∉	∉	∉		∉	+	∉	∉	∉	∉	∉		∉
Ramirez et al.	HSV		++	+	++	++	++	+	++		∉	+	∉	∉	∉	∉	∉		∉
Robertson and OCallaghan	sRGB		∉	Ø	∉	∉	∉	∉	∉		∉	Ø	∉	∉	∉	∉	∉		∉
Simula and Alhoniemi	HSV		∉	∉	∉	∉	∉	∉	∉		++	∉	∉	∉	∉	∉	∉		∉
Mittelstädt et al.	L*a*b		++	+	++	++	++	+	++		−	+	++	−	++	+	++		−
TeulingFig2	sRGB		∉	Ø	∉	∉	∉	∉	∉		Ø	Ø	++	+	++	+	++		+
TeulingFig3	sRGB		++	++	++	++	++	++	++		∉	++	∉	∉	∉	∉	∉		∉
TeulingFig3NoWhitening	sRGB		+	Ø	+	+	+	Ø	+		++	Ø	++	++	++	+	+		++
TeulingFig4a	sRGB		++	−	Ø	++	Ø	Ø	+		++	−	+	++	+	Ø	+		+
Wainer and Francolini	sRGB		−	Ø	∉	∉	∉	∉	∉		Ø	Ø	∉	∉	∉	∉	∉		Ø
Yeo et al.	sRGB		−	−	Ø	−	−	−	Ø		Ø	−	−	−	−	−	Ø		−

Table 3. Applicability of static 2D colormaps for analytical tasks, for black and white background colors. We suggest combinations of quality measures (cf. Tab. 2) for every task. The Borda count election method provides the overall scores aggregated to ++, +, Ø, and − in the order of applicability. We exclude colormaps if outnumbered by more than 80% of the competitors in one relevant metric ∉.

Figure 2.33: Image taken from the collaborative work with Bernard *et al.* [1]. *Quality assessment of state-of-the-art two-dimensional colormaps of Bernard et al. [1] who measured the performance of colormaps for visual analysis tasks. The metrics are similar to the metrics in this thesis and are based on the requirements discussed in this thesis. The static colormap provided in this thesis is among the most effective colormaps (see p. 93).*

We use cyan instead of green in order to approximately equalize the perceptually distance between all corner colors. The corner colors are equalized in intensity and saturation in the HSI color space (Keim, 2000) and then interpolated in the CIELAB color space.

The quality assessment of this colormap revealed that it provides a good trade-off for all tasks and is among the most effective colormaps (Figure 2.33). We apply this colormap in the user study of this section as state-of-the-art CIELAB colormap in order to judge the novel optimization method. The state-of-the-art RGB colormap has the same properties but was interpolated in RGB.

Experimental Design.

We conducted a user study with 8 visualization and data analysis experts. The study was within-subject designed. Each participant performed 18 tasks with each color mapping. The order of color mappings was randomized. The data was created according the experiment-algorithm of Ware and Beatty (1988), with the number of clusters (1 to 6 clusters), number of cluster elements (min: 30, max: 80), cluster positions and cluster shapes being randomized in each trial.

Results and Discussion.

The summary of results is illustrated in Figure 2.34. With our methods, which preserve clusters (2D and 3D version), users were significantly more accurate than with all other mappings on estimating the correct number of clusters (paired U-Test: $p<0.001$). This method supports the synoptic localization and comparison task of clusters and still preserves the local data distances. The configuration implies that clusters are known *a priori*, which is typically not the case in the synoptic identification task. However, this shows the advantage of

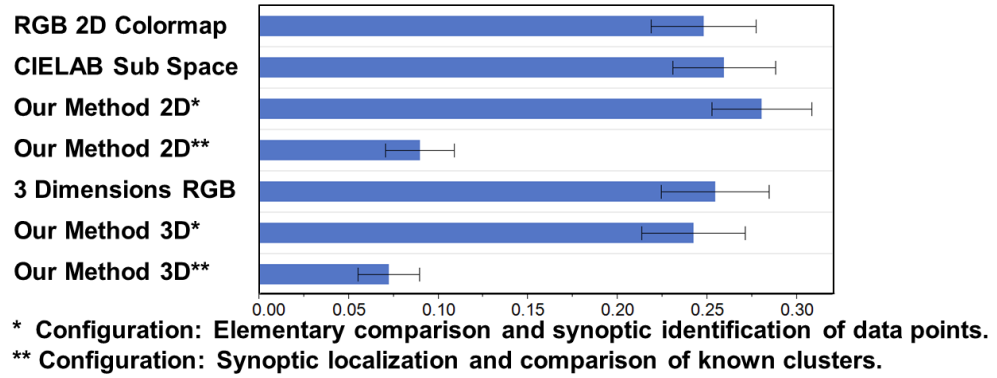


Figure 2.34: Image taken from [8]. *Evaluation Results. Averaged normalized error* ($|\frac{\#userEstimate}{\#clusters} - 1|$) *and standard deviation.*

concerning separation of known clusters in the color mapping.

Our method for synoptic identification provides perceptually correct mappings. The 3D version performed better, however, not significantly better than the state-of-the-art methods. The effect of perceiving more clusters if few are present (Ware and Beatty, 1988) seems to compensate the benefits of perceptual linearity. Especially, since our method tries to exploit the whole color space and preserves all pairwise distances.

We presented cost functions that are designed to support two opposing groups of analysis tasks. We argue that these functions are a sound basis for the analysis in realistic scenarios. However, we see the need for further research to support different analysis tasks and to improve visual cluster identification. It will be interesting to find trade offs in real applications. Further, we see future work to estimate the benefit of preserving global cluster relations and local cluster element relations in comparison to categorical color mapping.

2.4.5 Use Cases

This use case, text, and images are taken from the collaborative work with Steiger et al. [21]. For the division of responsibilities and work in this pub-

lication, please refer to Section 1.5 (p. 18). Text of other authors are formatted in italics and cited accordingly.

"We present a system to analyze time-series data in sensor networks. Our approach supports exploratory tasks for the comparison of univariate, geo-referenced sensor data, in particular for anomaly detection. We split the recordings into fixed-length patterns and show them in order to compare them over time and space using two linked views. Apart from geo-based comparison across sensors we also support different temporal patterns to discover seasonal effects, anomalies and periodicities. The methods we use are best practices in the information visualization domain. They cover the daily, the weekly and seasonal patterns of the data. Daily patterns can be analyzed in a clustering-based view, weekly patterns in a calendar-based view and seasonal patters in a projection-based view." (Steiger et al. [21])

"In this example, the operators in the control rooms are interested in the recordings of the power consumption. They are interested if sensors at two stations that are connected by electric cables measure similar values or not. Depending on the task, the network analyst wants to know about daily, weekly and seasonal patterns and trends. In how far do the consumption patterns change over the year? What are the differences between workdays and weekends? What are the regional differences in the grid?" (Steiger et al. [21])

"We contribute a visualization system that is able to assist the analyst in dealing with these problems. It consists of two tightly coupled views that complement each other: a Similarity View and a Network View (see Figure 2.35). *A topological map of the network gives a geo-based topological overview on the network in space and development of patterns over time for every sensor. Using a calendar-based visualization, the analyst is able to identify trends on different scales, based on individual sensors. As a result, the user can identify erroneous and suspicious measurements in the network. A similarity-based view gives important details on the global relations of different temporal patterns (in our example the power consumption over the day). The user can thus analyze daily patterns of the sensors, grouped by their pair-wise similarity. On demand, points that belong to the same sensor can be connected. This gives*

the analyst a quick overview on the variability of daily patterns over a period of time. If the patterns are very similar this spline would look like a tiny hairball and anomalies can be easily spotted. Tight linking between the two ensures that the user recognizes the same element and sets of similar elements in both views." (Steiger et al. [21])

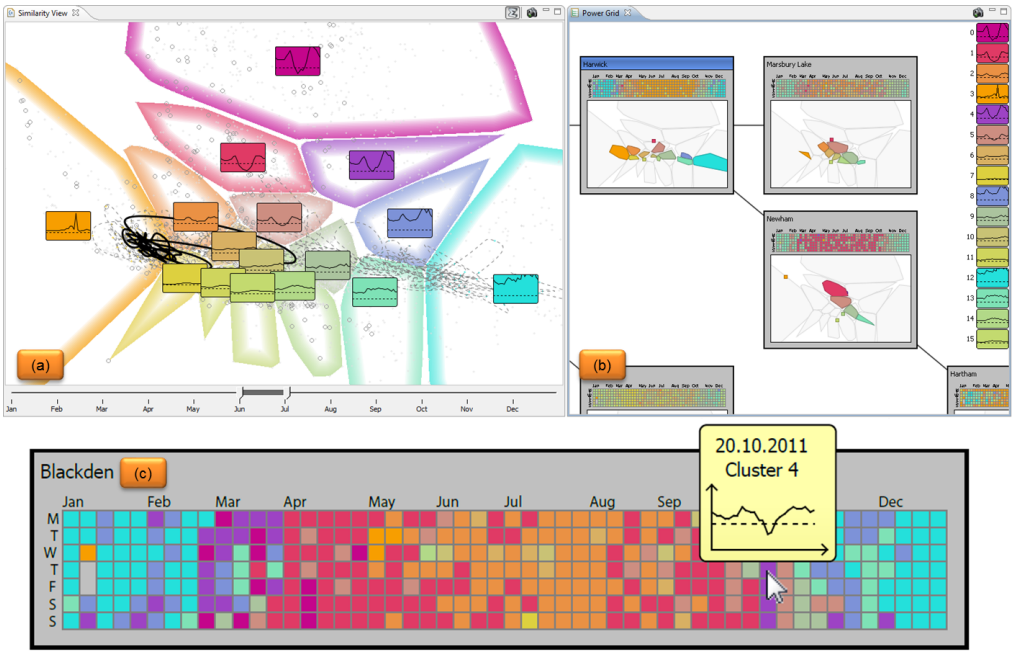


Figure 2.35: *Image adapted from the collaborative work with Steiger et al. [21]. "(a) The similarity view on the left side shows all daily patterns of all sensors. Similar patterns are assigned to the same group and color.* Each group is represented by a representative line chart colored in the group color. *The change of patterns over time for a selected sensor is indicated by the black spline. (b) The network view on the right side gives an overview of the network topology. The small calendar in the node glyph shows changes over time and a fingerprint view underneath shows the sensor patterns in the global context. (c) shows the calendar view in detail." (Steiger et al. [21])*

The similarity view can be created by projecting the daily patterns of measurement stations to a two-dimensional space and visualizing these two dimensions as spatial dimensions in a scatterplot. Thus, similar daily patterns share similar spatial locations and different patterns are separated. The projected data points are partitioned to identify cluster representatives. The color of the repre-

sentatives are assigned by projecting the spatial locations to color. Thus, similar colors encode similar daily patterns. This can be done with our method to encode high-dimensional data relations with perceptually equivalent colors or by state-of-the-art two-dimensional colormaps (Section 2.4.4 (p. 93)). In the similarity view, the user can, thereby, explore the different daily trends over the whole region and selected time frame (e.g., one year). In the upper part of Figure 2.35 (a), patterns with negative power consumption are visible. These patterns occur if the photo-voltaic energy production of the region produces more energy than it consumes. Figure 2.35 (b) and (c) shows the calender-view representations of different stations. Days in the calender are encolored with the color of the representative daily pattern of the similarity view. Therefore, color links the two visualizations and allows identifying interesting daily patterns as well as the stations, locations, and days, where such patterns occur.

2.5 Discussion and Future Work

In this chapter, we introduce color mapping requirements and quality metrics based on perceptual foundations for color encodings of single data dimensions and data relations for elementary and synoptic analysis tasks and their combinations.

We introduce an approach to generate (also color-blind safe) colormaps for each task combination and provide *ColorCAT*, which comprises the automatic requirement analysis for task-based color design for single data dimensions and interactively guides designers through the process of designing colormaps.

Further, we presented a method to project high dimensional data to perceptual uniform color spaces. Our method preserves the relationships of data items and supports the user in recognizing clusters while maximizing the exploitation of perceptual uniform color spaces. We provide guidelines on how to configure our methods for different analysis tasks and evaluated different versions of our

methods empirically. The results show that our method outperforms the state-of-the-art. However, we see some room for improvement and many further research challenges.

Integration of the Expert in the Design Process

Especially for categorical and ordinal data we suggest to use the brilliant colormaps of ColorBrewer (Harrower and Brewer, 2003). However, also ColorBrewer has limitations such as the limit of the number of colors and that it does not support (combined) analysis tasks. Further, there exist sophisticated color mapping algorithms that outperform the colormaps of *ColorCAT* in single analysis tasks or data types and even our own perceptual metrics could be used to provide optimal colormaps by optimization. Note, that *ColorCAT* always provides optimized colormaps, as long as the expert does not adapt the intensity or saturation features of the colormap.

However, the advantage of *ColorCAT* is that it also integrates the visualization expert in the design of colormaps to enhance the *semantic consistency* of colormaps. Because the expert is enabled to combine different analysis tasks and to modify the colormap to match the target domain, user preferences, and culture, which is not possible for automatic methods. The important steps of the color mapping algorithm are visualized and are, therefore, comprehensible to the designer who can modify the colormap by interactively changing color hues and adjusting properties on-the-fly. We argue that the integration of the visualization expert is very important for design processes of visualizations since the challenge of visualization design is to match the mental model of the target user, which cannot be captured by automatic methods.

Pre-attentive Identification of Clusters with Color

Our method, projecting high dimensional data into perceptual uniform color spaces, outperforms the state-of-the-art for encoding data relations, especially in the synoptic localization task. However, our results show that further research is required to improve synoptic identification with color (e.g., identifying clusters of color encoded data values) since the effect of perceiving more clusters

if few are present (Ware and Beatty, 1988) seems to compensate the benefits of perceptual uniformity. We were only able to provide a heuristic to overcome this effects for the localization tasks (if clusters and classes are known and should be preserved). The causes for these effects remain, however, unclear. Therefore, we see an open research gap for color encoding of data relations to explain and overcome these effects.

Data Transformation

There remains the need for engineering as well as constraints a designer encounters using colormaps of *ColorCAT*. Some data sets require normalization. For example, a data set with values in the range 0–100 and an outlier at the value 10.000 is badly represented with a standard min-max normalization and a linear colormap since the analyst would not perceive the variation in the data range. Therefore, either the colormap or the data values must be transformed. If the colormap is perceptual linear, it is guaranteed that the analyst will perceive the data normalization as intended, which would not be the case if also the colormap was distorted by a transformation. Any normalization must be applied in the data space since the colormaps of *ColorCAT* aim for perceptual linearity.

Another challenge is the support of segmented quantitative data. In contrast to ordinal data, segmented quantitative data requires that segments are ordered and perceptually distinct but the continuous variation within each segment must be encoded. A first approach to this issue was presented by Tominski et al. (2008).

Volume Renderings & Spatial Frequency

Our guidelines are valid for abstract (two-dimensional) data visualizations. Volume renderings or visualizations of natural forms, streams, and motion typically require luminance, saturation, and transparency for representing depth or motion. There is evidence that fine details (high frequency patterns) are carried effectively by the luminance (achromatic) channel (Ware, 2012). Mullen (1985) found that the chromatic channels only carry one-third of the amount of detail compared to luminance and are, therefore, not recommended for high spa-

tial frequency data. Guidelines concerning this issue are discussed in detail by Bergman et al. (1995) and Rogowitz et al. (1996) and implemented in the PRAVDA system (Rogowitz and Treinish, 1993). This issue is very important for the task of perceiving natural forms and shapes, which should, therefore, be supported by luminance.

However, the perception of fine details is effective because our eye amplifies local high-frequency details to a significant extend. Thus, we perceive these details due to contrast effects, which is harmful for most analysis task (other than perceiving natural forms and shapes).

Since the luminance channel is already used in our guidelines to encode information, our methods are to a certain extend not applicable to these applications. However, the requirements of the analysis tasks remain the same. It would be interesting to extend our guidelines to the application domain of volume renderings and visualizations of natural forms and streams.

Similar to this thesis, requirements, metrics, and guidelines could be defined to support analysis tasks in this domain, which would increase the effectiveness of the renderings significantly since the analyst is explicitly supported in the target analysis task.

Colormaps on Complex Backgrounds

In geo-spatial data analysis, it is important to visualize information in the geo-spatial context. Therefore, color encoded data is often layered on top of cartographic maps in this application domain. This is a challenging design problem since cartographic maps often use color encodings to represent cartographic features, which should not interfere with the colormap that encodes the additional data. Harrower and Brewer (2003) provide guidelines for different data types in this application domain, however, do only support few color levels.

It depends on the analysis task and the priority of the cartographic features on how colors could be used to encode additional information on maps. For example, if the task is focused on the identification or localization of data values, the cartographic features should only be encoded with a grayscale and the data with multiple color hues with maximized but equalized saturation and intensity.

Thereby, the analyst can effectively identify and localize data values but still perceive the cartographic features without interference.

However, if the color of cartographic features cannot be removed, this is not a valid option. Another possibility is the inversion: equalizing brightness and saturation of map colors and layering the data on top with a grayscale. This allows effective comparison of data values and representation of cartographic features. Ware (2012) provides further guidelines for this use case, nevertheless, complex backgrounds or multiple layers of data is one of the most challenging tasks for the design of color encodings and few works have been done for this research gap.

Ahestetic Design Goals

Aesthetic design goals are also very important in colormap design, because aesthetic designs reduce the stress of visual analysis tasks (Wang et al., 2008) and make the use of tools more enjoyable (Norman, 2002). It remains an open question how to satisfy perceptually-motivated metrics and to allow enough artistic freedom for colormap design simultaneously. Novel colormaps are presented by Samsel et al. (2015) who derived the designs with trained artists. It would be interesting to extend the methods in this chapter into this direction.

Semantic Support

The challenge of *semantic consistency* could only implicitly satisfied by integrating the visualization expert in the design process to adapt the colormap to the domain, culture, and preferences of the target user. This is a first step, however, in order to directly integrate semantic meaningful and consistent colors, we see a general challenge for the research field. In Section 6.2 (p. 225) “Color Anchors”, we explicitly discuss more details and how this challenge could be tackled.

Output Device

The methods presented in this chapter produce color encodings for standard LCD displays (calibrated to sRGB (Anderson et al., 1996)). Other output devices such as beamers and printers were left out. Due to hardware profiles and standardized color spaces, colors should be transferable from device to device but practically this is not the case. Section 4.4.3 (p. 166) discusses methods to calibrate output devices and provides a method that compensates for different ambient lights and display primaries. However, most of the existing methods are based on *gamma correction* for calibrating output devices and renderings. They cannot measure and account for gamuts of output devices that bend perceptual uniform color spaces. Thus, guidelines, designs, and algorithms based on perceptual linearity are biased. A first promising method was introduced recently by Flatla and Gutwin (2011) who provide methods to measure individual characteristics of color vision deficiency by showing samples to the target user and measure his/her perception by interaction. We see high potential in adapting this method to measure (malformed) gamuts of output devices and to adapt color mappings accordingly. However, to design adaptive colormaps to the output device remains one of the most interesting and important research challenges.

3

Compensation of Contrast Effects

Contents

This chapter is based on the following publications and major parts of the sections also appeared in the following publications:

[14] Sebastian Mittelstädt, Andreas Stoffel, and Daniel A. Keim. *Methods for Compensating Contrast Effects in Information Visualization*. Computer Graphics Forum, 33(3):231–240, 2014.

[11] Sebastian Mittelstädt and Daniel A. Keim. *Efficient Contrast Effect Compensation with Personalized Perception Models*. Computer Graphics Forum, 34(3):211–220, 2015.

For the division of responsibilities and work, as well as a statement of contributions in these publications, please refer to Section 1.5 (p. 18).

Chapter Abtract & Structure

Effective color encoding requires that the mapping is *faithful*, *expressive*, *pre-attentive*, *semantic consistent*, and *supports the analysis task*.

Chapter 2 provides guidelines to design effective color encodings. However, the challenge of *faithfulness* can only partially be satisfied by colormap design, since optical illusions such as contrast effects significantly bias our perception of colors. For instance, a gray patch appears brighter on a black background than on a white background (Figure 3.1). How a pixel is perceived in the visualization can only be estimated in the final rendering and cannot be overcome in the design phase. Accordingly, the perception of color-encoded data items depends on their surround in the rendered visualization. Optical illusions are, therefore, considered harmful in data visualizations.

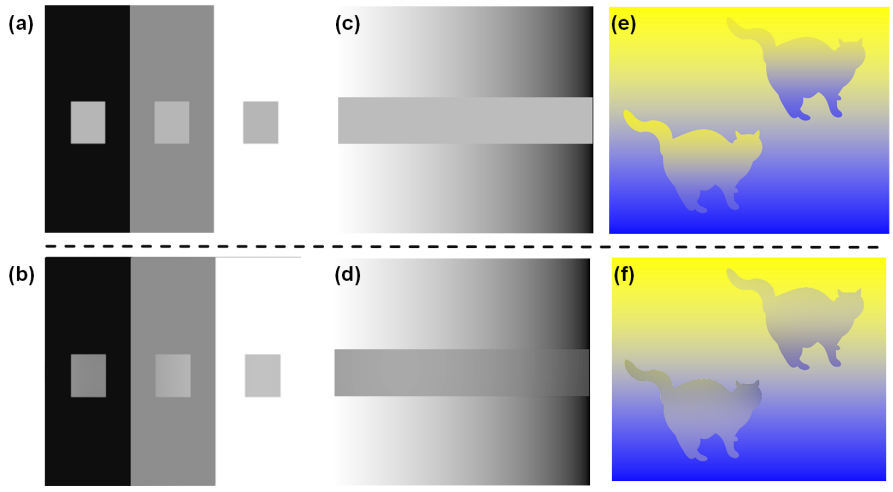


Figure 3.1: Image taken from [14]. *Standard examples of simultaneous contrast effects.* ***(Top)*** *Original images. The gray patches share the same gray values but are perceived differently* ***(a)*** *or as a gradient* ***(c)****. The cats* ***(e)*** *share the same gradient from less saturated blue to yellow.* ***(Bottom)*** *Compensated images. The patches and cats are almost perceived equal.* ***(b)*** *Our method reduces the contrast effects on the patches to faithfully represent the gray value (Eq. 3.3).* ***(d)*** *The method reduces the perceptually induced gradient (Eq. 3.6).* ***(f)*** *Our method preserves the global gradient and average color of the cats (Eq. 3.7).*

Studies of Cleveland and McGill (1983), Ware (1988), and Brewer (1996b) have shown that contrast effects have significant impact on the accuracy of users reading and comparing color encoded data. Even if we are aware of these perceptual issues, we cannot compensate cognitively for these effects because they are pre-attentively processed.

To overcome this limitation, we present a methodology and method based on perceptual metrics and color perception models to reduce physiological bias such as contrast effects. We evaluate our technique with a user study and find that the technique doubles the accuracy of users reading and comparing color encoded data values. Since the presented technique can be used in any application without adaption to the visualization itself, we are able to demonstrate its effectiveness on data visualizations in different domains.

This chapter describes the general methodology of contrast effect compensation and provides an effective method that is evaluated in an extensive user study with 40 participants. In Chapter 4, the methodology is extended by efficient algorithms for interactive visualizations but also to provide methods for personalizing perception models to capture individual contrast perception and different environmental effects. The methods provided in this chapter build, furthermore, the base for Chapter 5, which exploits contrast effects to enrich visualizations by visual boosting.

In this chapter, the reader learns about the impact of contrast effects in data visualization in Section 3.1. Related work is discussed in Section 3.2, that provides predominantly rules-of-thumbs or handcrafted color sets that reduce the bias of contrast effects. However, they are not able to compensate for contrast effects. Section 3.3 introduces how to measure the impact of physiological biases and how to define the compensation of contrast effects as an optimization problem. Further, Section 3.3 provides the methodology and method to compensate for contrast effects, which is evaluated in two experiments with 40 participants in Section 3.4. Use cases are provided in Section 3.5. The chapter is concluded with a discussion and future work in Section 3.6.

3.1 The Impact of Contrast Effects on Visual Data Analysis

Optical illusions may bias the analysis process at perceptual and cognitive levels. In our cognition, illusions are caused by assumptions about the relation of visual objects which lead to unconscious inference. Perceptual illusions are caused by the processing of physical stimuli. Incoming light is first processed by nerve cells that measure the intensity of light of different wavelengths (short, mid and long wavelengths). This information is passed to cells that combine this information to perceive luminance and color. In order to detect and identify natural shapes, objects, and movement our visual perception has evolved to be good at detecting edges and determining different color hues. Amplifying contrasts is a means of achieving this.

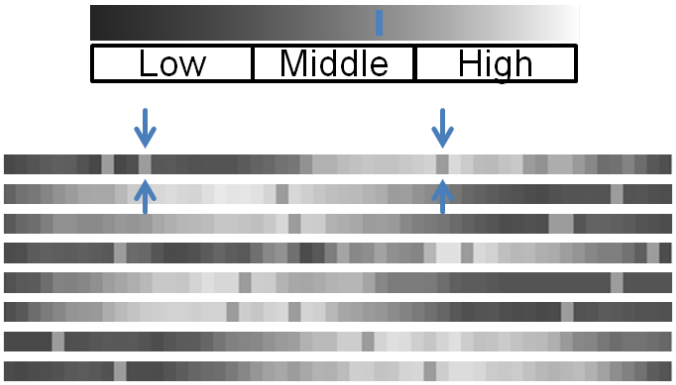


Figure 3.2: Image taken from [14]. *Pixel-cell based visualization for time series data. The highlighted values share the same data value of* 0.6. *Due to the surrounding regions in the visualization, contrast effects let us perceive the data points differently.*

In data visualizations, these contrast effects can be harmful. Ware (1988) has quantified the bias of contrast effects, concluding that contrast effects can cause errors of up to 20 %, leading to analysts grouping wrong data points and

detecting relations or extremes that do not exist in the data. Figures 3.1 and 3.2 demonstrate the extent of these contrast effects. As with most optical illusions, humans cannot compensate for these effects even if they are aware that the representations are wrong.

There are sophisticated guidelines and discussions of how to design colormaps to avoid contrast effects. However, color perception is relative. The perceived color of an area of a display can only be predicted, if the surrounding colors are known. Thus, these guidelines cannot guarantee unbiased displays.

3.1.1 Contributions

In this section, we present the first post-processing method for compensating contrast effects in visualizations as illustrated by the standard examples in Figure 3.1. We claim the following contributions:

1. A methodology and method for compensating physiological bias such as contrast effects based on color appearance models and optimization algorithms that can be used on any data visualization as a post-processing step.

2. A definition of the optimization goal for contrast effect compensation and the corresponding perceptual metrics.

3. A general heuristic to approximate the gradient of compensation.

4. An empirical evaluation with 40 participants of the perception model and the compensation, based on realistic tasks and data.

While computational models for physiological effects exist, cognitive effects are far more complex and concrete theories or models have yet to be formulated. We, therefore, exclude cognitive effects from the scope of this thesis.

3.2 Related Work

The influence of contrast effects on data visualizations has been quantified and evaluated by Cleveland and McGill (1983), Ware (1988), and Brewer (1997a), who indicate that their presence significantly influences users reading visualizations. In the following, some strategies have been proposed to handle these effects.

Ware found that the metric task (reading metric quantities) is significantly influenced by contrast effects, when using a color scale along one (a)chromatic channel. Some solutions (Ware (1988), Levkowitz and Herman (1992), Keim (2000), and Kindlmann et al. (2002)), therefore, propose *spiral* colormaps, which maximize color differences by varying over hues with linear increasing intensity, in order to create colormaps that alternate between our chromatic channels, thereby reducing the probability of contrast effects.

A method for predicting simultaneous contrast on maps was presented by Brewer (1996b) – the models and guidelines are used to create sets of colors, for which the probability is reduced that they influence the user by contrast effects in the final visualization.

However, the perception of colors on the display is not taken into account in all the listed techniques above. Without knowing the surrounding color of a data point, it is impossible to accurately estimate its perception *a priori*. We, therefore, propose a post-processing method to cope with these effects. For this goal, the appearance of colors needs to be predicted.

How color is perceived and how to model illusions is still a focus of research but appropriate models exist. Hunt and Pointer (2011) and Fairchild (2013) offer a full discussion of color appearance. The standardized color appearance model CIECAM02 (Moroney et al., 2002) is based on the results of the Hunt model but is adapted for industrial practical usage. These models are still based on certain viewing conditions and fixed patch sizes. Further advances for more

complex stimuli of natural scenes and varying viewing conditions have been made in the iCAM framework by Fairchild and Johnson (2004).

Many computational models for brightness effects have been developed over the last decades. For instance, the anchoring theory of Gilchrist et al. (1999) and the Scission theory of Anderson and Winawer (2005) try to explain these effects on high-level (cognitive) interpretation of a scene by segmenting and processing the complex scene into frameworks and layers. In contrast, low-level models such as the ODOG model of Blakeslee et al. (2005) successfully use simple Gaussian based convolutions to predict a variety of contrast effects such as simultaneous contrast, Mach bands, Hermann grid and White's illusion. A full introduction and discussion of achromatic vision can be found in Gilchrist's book (Gilchrist, 2006).

3.3 Method for Compensating Contrast Effects

Our methodology to compensate physiological biases in data visualizations is an optimization process. The goal is to determine an image I' that is perceived as the original image I and faithfully represents the data. Our method determines I' in an iterative process illustrated in Figure 3.3: First, the method estimates in step t the bias for a given input image I^t at each pixel I^t_p. Then these effects are compensated by changing the color of pixel I^t_p and its surround S^t_p to reduce the bias. These steps are iteratively performed until the bias for all pixels is compensated. The bias at each pixel is estimated by a perception model PM that predicts the perceived image P^t of I^t. The data is faithfully represented if the perceived image $PM(I^t, S^t) \mapsto P^t$ is equal to image I. The goal of compensation can thereby be defined by minimizing the difference of the perceived

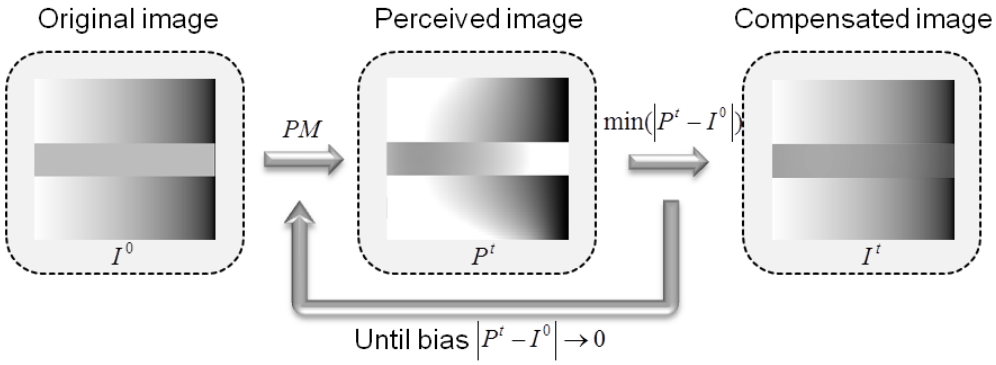


Figure 3.3: Image taken from [14]. *Schematic approach. The method iteratively reduces the difference between perceived and original image.*

image and the original image (Eq. 3.1).

$$\min\left(|PM(I,S) - I|\right) \tag{3.1}$$

A sound perception model PM is, therefore, the basis of the compensation method. Since vision research still advances to optimize or create new models to cover all physiological effects and some visualization techniques may suffer under different effects, we propose a methodology, where the perception model is an updateable module.

If the model PM is homogeneous and invertible, then the effects on one pixel I_p can be inverted and the color of pixel I'_p, which is correctly perceived, can be estimated. Another solution is to change the surrounding pixels S_p in order to faithfully represent I_p. Both solutions work when considering a solitary pixel, but in a multi-pixel display the adjustment of one pixel will change the perception of other pixels. Therefore, the compensation method must find an optimum to compensate the effects for the whole visualization. Our process, therefore, requires perceptual metrics as cost functions to evaluate a solution I^t, a sound perception model PM, and an optimization algorithm to find solutions that meet the compensation goal (Eq. 3.1).

3.3.1 Estimating Physiological Bias in Visualizations

In the following sections, we define our cost functions based on perceptual metrics and different visualization tasks as defined in Section 1.6.2 (p. 27). The visualization tasks are grouped in two levels: elementary and synoptic tasks. While elementary tasks address individual data points, synoptic tasks consider sets of values or data points.

Color Difference Measure

Perceptual color distances are the base of all our perceptual metrics. We use the DIN99 color space (Cui et al., 2002) for distance measurements. This color space is an extension to CIELAB that accurately models small color distances within $\Delta E < 5$ and allows vector arithmetics (see also Section 1.6.1 (p. 24)).

$$\Delta E(c_1, c_2) = \sqrt{\Delta L^2 + \Delta a^2 + \Delta b^2} \tag{3.2}$$

Cost Functions for the Elementary Tasks

Any difference between the perceived colors of data points and the original color of data points misleads users in the localization, identification and comparison tasks. Therefore, the costs have to increase if a data point is perceived differently as the original color. While the foreground F that holds the data must be accurately visualized, the background B can be used as an additional space to reduce the effects on the foreground. We, therefore, differentiate between foreground and background.

The first measure f_1 (Eq. 3.3) computes the difference between the original color and the perceived color of each pixel in the foreground (F). Respectively, measure f_2 (Eq. 3.4) computes the difference for the background (B) in order to avoid disturbing halos or artifacts. Figure 3.4 shows the result of the method focusing on preserving the color of each individual pixel p to minimize f_1. In

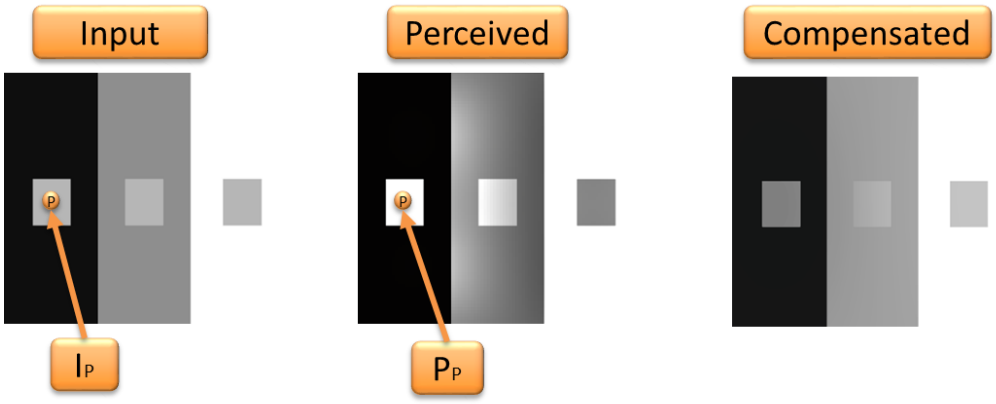


Figure 3.4: *Preserving single pixel values. The three gray patches in the input image share the same color. The black, gray and white background lead to biased perception caused by contrast effects. This is compensated in the right image by focusing on* f_1.

order to avoid misinterpretation of the background, the cost function f_3 (Eq. 3.5) measures the distance of a perceived background pixel color P_p to the closest color of the color map $CM(P_p)$. If the distance is very close ($\Delta E < 5$), users may interpret the background as a data point, which harms identification and localization of data values.

$$f_1(I,P) := \frac{1}{|F|}\sum_{p\in F}\Delta E(P_p, I_p) \tag{3.3}$$

$$f_2(I,P) := \frac{1}{|B|}\sum_{p\in B}\Delta E(P_p, I_p) \tag{3.4}$$

$$f_3(I,P) := \frac{1}{|B|}\sum_{p\in B}\max(0,\ 5 - \Delta E(P_p, CM(P_p)) \tag{3.5}$$

Cost Functions for the Synoptic Tasks

The faithful representation of single data values is ensured by the measures above. However, these measures do not preserve local and global relationships. To preserve local distances between each pixel P and its neighbors N and thereby spatially connected structures, we apply the local measure f_4 (Eq. 3.6). It effectively removes the bias of perceived (but not real) gradients (Figure 3.5).

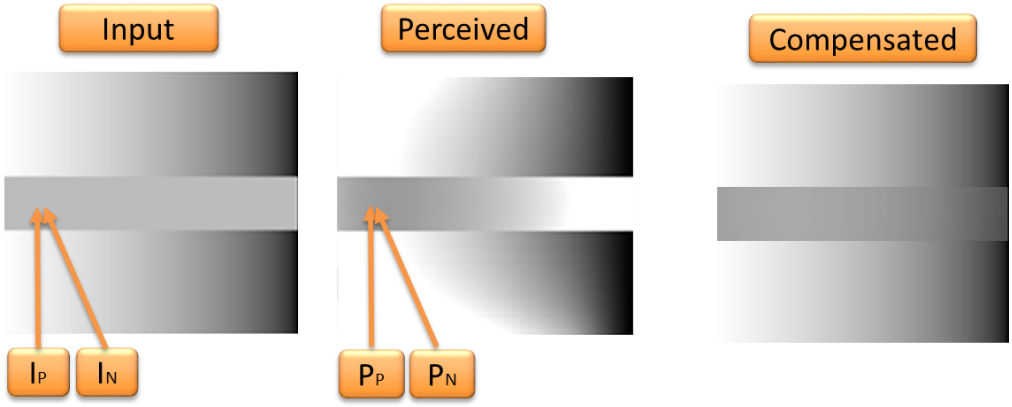


Figure 3.5: *Preserving local neighborhoods. The gray bar in the input image has the same color over its lengths. Due to the color gradient of the background, we also a perceive gradient in the bar. The perceptual gradient is removed by focusing on* f_4.

An issue with preserving global structures or color patterns is that they may not be spatially connected and not known *a priori*. This makes these patterns hard to preserve. Our method can, however, ensure that the global relations and the average impression between regions of the image are faithfully represented such as the cats in Figure 3.6. The constraint of faithfully representing each data point is, therefore, relaxed in order to preserve the global relationships of data items. This can be obtained by computing a Gaussian pyramid of the original and perceived images with $\delta = 2^\circ$ as first level according to the attentional center of the CIE standard observer (Eq. 3.7). Each pixel is compared to its according position in the pyramid levels. We decided to use a Gaussian pyramid, because it measures the perception of the images at different resolutions. Thereby, the measure accounts for structures of different granularity.

$$f_4(I,P) := \frac{1}{|F|}\sum_{p\in F}\sqrt{\frac{1}{|N_p|}\sum_{n\in N_p}(\Delta E(P_p,P_n)-\Delta E(I_p,I_n))^2} \tag{3.6}$$

$$f_5(I,P) := \frac{1}{|G|}\sum_{G_l\in G}\frac{1}{|F|}\sum_{p\in F}\Delta E(G_l(P)_p, G_l(I)_p) \tag{3.7}$$

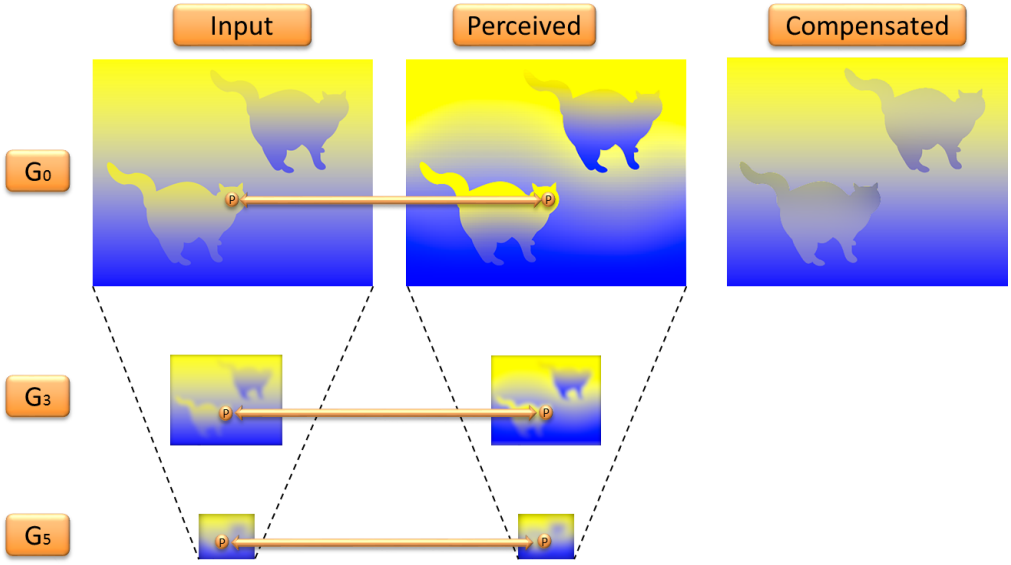


Figure 3.6: *Preserving structures of different granularity. The cats share the same (dull) gradient from yellow to blue. Due to the gradient in the background, we perceive the cats differently. The average color of the cats is preserved in the right image by* f_5.

Combinations of Cost Functions

Since all of the cost functions are based on the color distance measure, the cost functions do not have to be scaled and can be aggregated in a weighted sum:

$$f(I, I^t, PM) := \sum_{i=1}^{n=5} \lambda_i \cdot f_i(I, PM(I^t)) \tag{3.8}$$

The elementary identification and comparison of data values are the analysis tasks that suffer the most under the influence of contrast effects.

f_1 is the most important function since it preserves the color at each pixel and provides faithful identification of the encoded data values. Together with f_4, which aims to preserve the color difference to its neighbors, f_1 and f_4 can provide faithful visualizations. f_3 ensures that the background colors do not appear as data values, which could lead to confusion of the analyst. For typical data analysis scenarios (e.g., elementary identification and comparison of data

values), we propose to assign f_1, f_3, and f_4 high (equal) weights in order to provide faithful visualizations.

We assign f_2 a very low weight (in our examples and evaluation, we assigned zero) since this allows the background to compensate effects in the foreground. f_5 is a special cost functions that preserves the average colors of structures of different granularity and can be omitted if this is not required by the application.

In our experience and evaluation, equalizing the weights of all cost functions beside f_2 and f_5 provided satisfactory results (see also Section 3.3.4 (p. 124)).

3.3.2 Perception Model

The iCAM framework (Fairchild and Johnson, 2004) has been developed as an enhancement of the standardized CIECAM02 (Moroney et al., 2002) color appearance model for image processing. It is robust in predicting color appearance and contrast effects, such as simultaneous contrast and chroma chrispening (Figure 3.7).

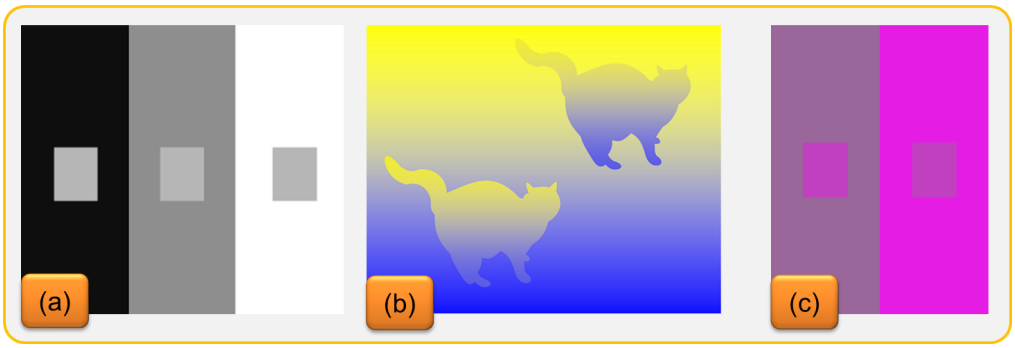


Figure 3.7: *Physiological effects predicted by iCAM (Fairchild and Johnson, 2004). The patches in each example share the same color. (a) Simultaneous contrast: The surrounds differs in lightness and let the patches appear brighter or darker than they actually are. (b) Color contrast: the surrounds of the cats differ in hue and saturation, which let the cats appear in different hues. (c) Chroma chrispening: The surrounds differ only in saturation, which let the left patch appear brighter and more vivid.*

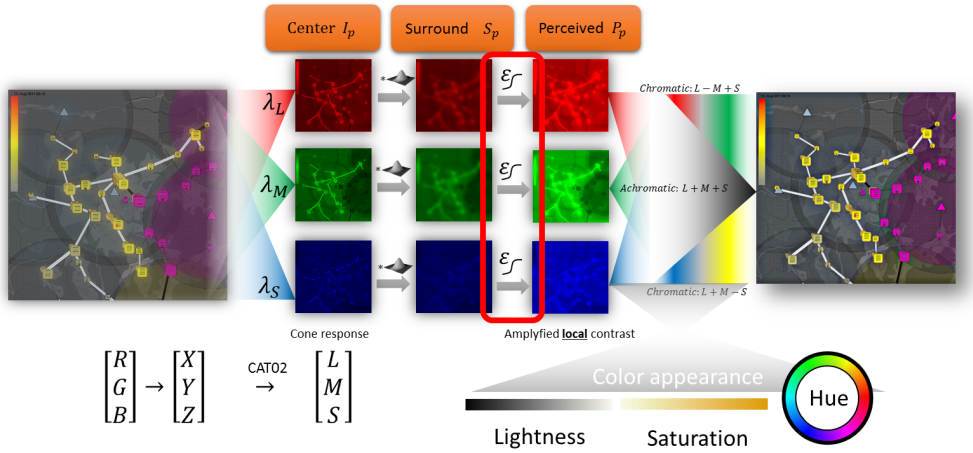


Figure 3.8: Image taken from [11]. *Perception model for predicting contrast effects according to the iCAM framework (Fairchild and Johnson, 2004). The original image is converted from RGB to CAT02 estimating the activation of the cones in the human eye for red, green, and blue light (LMS channels). The perception of the surround is modeled by Gaussian convolution. The eye naturally amplifies contrasts to boosts structural information to detect edges and natural shapes. This is done by amplifying local contrast by the cones that compare center and surround information, which is calculated by the model in each LMS channel. Thereby, the perception model predicts contrast effects on the lowest levels of color perception (LMS channels, representing the cones). The LMS channels are combined by higher processes to two opponent chromatic channels (red-green, blue-yellow) and one achromatic channel, which are the base for the color vision channels lightness, saturation, and hue (Section 2.3.2 (p. 45)).*

Revisiting the iCAM Framework

The iCAM framework (Fairchild and Johnson, 2004) processes the image in the following way (Figure 3.8): First, the image is separated into three intensity images according to the response signals of the cones in the human eye. The color of each pixel is, therefore, transformed from sRGB into the CAT02 color space (Moroney et al., 2002). This estimates the activation of the three cone types in the human eye, which process long, medium, or short-wavelength light (LMS channels). Second, each LMS channel (intensity image I) is convolved with a Gaussian kernel (Eq. 3.10) to model the perceived surround S_p of each pixel (with x, y being the pixel distance to pixel p, σ see below). Then, the model adapts each color at pixel p of image I to its surround S_p and to the

reference white $D65$ of sRGB (Eq. 3.9). These steps lead to a local chromatic adaptation, which models a variety of contrast effects (Figure 3.7).

$$PM(I, S, \epsilon) = \left(c_1 \cdot \frac{D65}{c_3 \cdot (S_p / I_p)^{\epsilon}} + c_2 \right) \cdot I_p \tag{3.9}$$

$$S_p(\sigma) = \sum_{x,y} I_{x,y} \cdot \frac{1}{2\pi\sigma^2} \cdot e^{-(x^2+y^2)/(2\sigma^2)} \text{with } x, y \in [-2\sigma, 2\sigma] \tag{3.10}$$

The reference white can be used to model ambient light or other lightening conditions. The parameter ϵ controls the extent of perceived local cone contrast. The higher ϵ the more contrast is reduced by our method. More details can be found elsewhere (Fairchild and Johnson (2004) and Moroney et al. (2002)) or in Chapter 4. The constants are selected according to Fairchild and Johnson (2005) with $c_1 = 0.94$, $c_2 = 0.06$, and $c_3 =$ the maximum of the LMS channel $[94.92, 103.54, 108, 74]$ in the CAT02 color space.

Our Extensions to the iCAM Model

The size of the kernel that models the surround appearance is a critical parameter. The kernel size σ is estimated in iCAM by 20% of the image width. According to Shi et al. (2013) and our study, we found that this is insufficient since simultaneous contrasts are increased in images with high spatial frequency. We, therefore, adjust the kernel to the spatial frequency of the image. We filter the image by difference-of-Gaussians of varying sizes ($\sigma_1(i) = 2^i$ and $\sigma_2(i) = 1.6 \cdot \sigma_1(i)$) with increasing i, which accords human perception of edges in natural scenes (Marr and Hildreth (1980) and Young (1987)). The root-mean-square response of each filter (Eq. 3.11) indicates the presence of the according frequencies. The filter sizes are then pooled according to their responses, which approximates the average kernel size that will consider most of the spatial frequencies of the image.

The second critical parameter is the exponent that steers the adaption. There is a difference between bright centers with dark surrounds and dark centers with bright surrounds, as reported by Blakeslee and McCourt (1999). We, therefore,

differentiate between these cases (Eq. 3.12). In our experiments, we found that $\epsilon_0 = 0.5$ (adapted from (Fairchild and Johnson, 2005)) and $\epsilon_1 = 0.6$ provide satisfactory results. These parameters are discussed in detail in Chapter 4.

$$RMSE(i) := \sqrt{\frac{1}{|p|}\sum_{p}\left(S_p(\sigma_1(i)) - S_p(\sigma_2(i))\right)^2} \tag{3.11}$$

$$PM'(I,S) := PM(I,S,\epsilon), \text{ with } \begin{cases} \epsilon = \epsilon_0, \text{ if } S_p > I_p \\ \epsilon = \epsilon_1, \text{ else.} \end{cases} \tag{3.12}$$

3.3.3 Optimization Algorithms & Heuristics

The optimization goal (Eq. 3.1 (p. 113)) can be achieved by minimizing the sum of cost functions (Eq. 3.8 (p. 117)). The problem space is non-linear and may be non-convex with non-linear constraints depending on the perception model. The model itself may be continuous, differentiable, and homogeneous. These properties significantly influence the selection of an efficient optimization algorithm. Since we do have a meaningful initialization for the optimization (the original image) and a rough approximation of the gradient of compensation (see below), heuristics have great potential.

One of the most effective group of methods in numerical optimization are gradient methods. By inverting the perception model, the gradient of compensation can be estimated. Some models, such as our model, are not differentiable and thus, the gradient can only be approximated. This can be revealed by determining the direction (vector in the DIN99 color space) of contrast effects.

For example, a bright patch of gray is perceived even brighter if the surrounding is dark. Thus, the perception model predicts an increase of intensity for the patch and a decrease of intensity for the surrounding. In order to compensate this effect, either the patch itself has to be made darker to reduce the effects on itself or the surrounding has to be made brighter to compensate the effects on the patch.

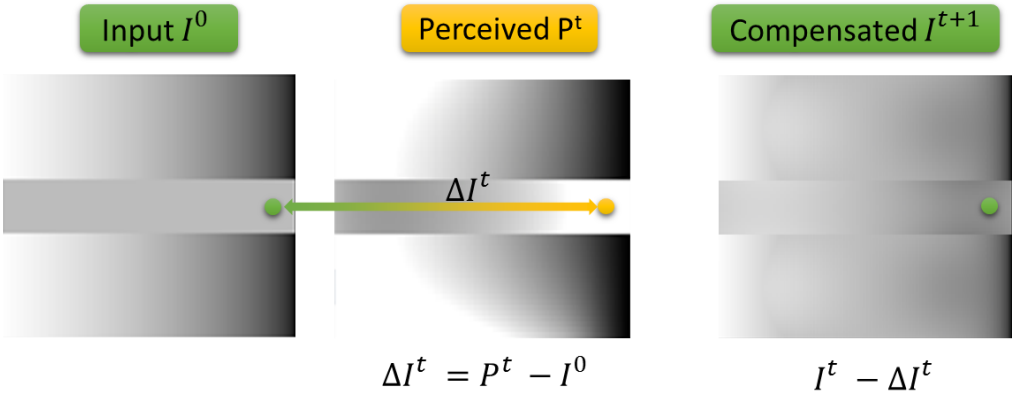


Figure 3.9: *The gradient of compensation can be approximated by the adaptation of the foreground. P^t is the perceived image of the image I^t in iteration t. The contrast effect ΔI^t on each foreground pixel is inverted and used to adapt the color of the pixel with $I^{t+1} = I^t - \Delta I^t$. This compensates the contrast effect for single and only for single pixels since changing one pixel will influence the perception of other pixels; therefore, this has to be done in several iterations.*

Foreground Adaptation

Our iterative heuristic approximates the final, compensated image I' in step t with I^t. It starts in step $t = 0$ with $I^0 = I$. The effects on one pixel ΔI_p^t can be estimated by calculating the difference of the perceived image $PM(I^t, S^t)$ to the original image I (Eq. 3.13) with S^t being the perceived surround and $\Delta I_p^t = 0$ for background pixels. If the model PM is roughly continuous and homogeneous, the effects will influence a pixel I_p^{t+1} in the same way. Therefore, the difference between perceived and original pixels can be reduced if, I_p^{t+1} is selected such that $I_p^{t+1} + \Delta I_p^t = I$. In the example, the bright patch will become darker (inverse direction of ΔI^t) in order to compensate the effects on itself (Figure 3.9).

$$\Delta I^t = PM(I^t, S^t) - I \tag{3.13}$$

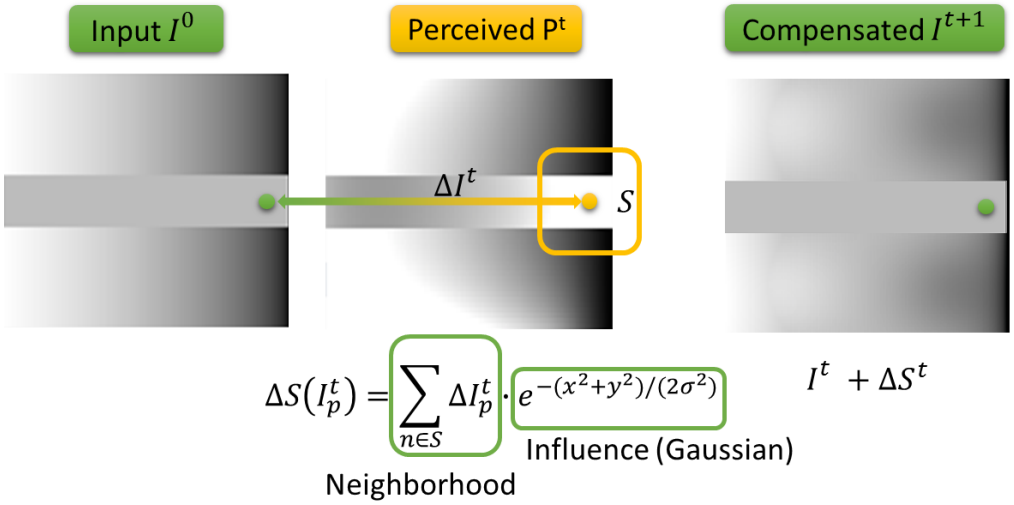


Figure 3.10: *The gradient of compensation can be approximated by the adaptation of the background. P^t is the perceived image of the image I^t in iteration t. The contrast effect ΔS^t on neighboring pixels are estimated and used to adapt the color of the pixel with $I^{t+1} = I^t + \Delta S^t$. This eliminates the contrast effects on the center pixel but changes the appearance of the surrounding pixels to a major extend. The method has to find a balance.*

Background Adaptation

A pixel I_p^t can also change its color to reduce the effects on other pixels. The pixel measures the effects $\Delta S(I_p^t)$ (Eq. 3.14) on its surround S_p given by PM. The effects on surround pixels are summed according to the weights of the model of the surround in the perception model (Gaussian kernel in iCAM). If pixel I_p^t adapts in the direction of $\Delta S(I_p^t)$, it will reduce the contrast effects for its surroundings. In the example, the dark surround will become brighter (direction of $\Delta S(I_p^t)$) in order to compensate the effects on the bright patch (Figure 3.10).

$$\Delta S(I_p^t) = \sum_{x,y} \Delta I_{x,y}^t \cdot \frac{1}{2\pi\sigma^2} \cdot e^{-(x^2+y^2)/(2\sigma^2)} \tag{3.14}$$

Gradient of Compensation

The two different approaches of compensation for single pixels can be combined to approximate the gradient ΔG of compensation (Eq. 3.16) by minimizing the combination of cost functions (Eq. 3.8) for all pixels. As described above, inversion of the perception model is not enough, since changing one pixel will influence its surrounding. We, therefore, calculate the compensated image I' with multiple iterations with Eq. 3.15 and control the step size of compensation with $\varphi_{1,2} \in [0, 1]$ (Eq. 3.8). The pixels of I^{t+1} will adapt to the change of their surround in I^t. In practice (see our instantiation in the next section), we set $\varphi_1 > 0, \varphi_2 = 0$ for foreground pixels and $\varphi_1 = 0, \varphi_2 > 0$ for background pixels. In this way, the background, which does not contain valuable data, compensates the effects on the foreground.

$$I_p^{t+1} = I_p^t + \Delta G_p \tag{3.15}$$

$$\Delta G_p = -\varphi_1 \cdot \Delta I_p^t + \varphi_2 \cdot \Delta S(I_p^t) \tag{3.16}$$

It should be noted that this approximation of the gradient is independent of the perception model. The required influence functions of the surround in Eq. 3.14 $e^{-(x^2+y^2)/(2\sigma^2)}$ can be replaced by other functions. See Section 3.6 (p. 137) for a discussion of applicable perception models in this methodology.

3.3.4 Instantiation of the Method

For our experiments and applications we use the following instantiation and parameters:

Parameterization

The perception model is parameterized as described in Section 3.3.2 (p. 120) with $\epsilon_0 = 0.5$, $\epsilon_1 = 0.6$. The step sizes of the iterations are set to $\varphi_1 = \alpha, \varphi_2 = 0$ for foreground pixels and $\varphi_1 = 0, \varphi_2 = \alpha$ for background pixels of the gradient G (Eq. 3.16) with step size α, which is the target parameter of optimization.

Our tasks are focused on elementary metric readings and synoptic identification of local neighborhood trends. Therefore, we prioritize cost functions f_1–f_4 by equalizing their weights in the iterative steps and exclude cost function f_5 that preserves global relationships. For the final decision, we select the optimum solution that minimizes all cost functions f_1–f_5.

Optimization

We apply *simulated annealing* (Kirkpatrick et al., 1983) with multiple threads. Each thread is initialized with the original image as starting point for the optimization. In each iteration, the neighbor solution is determined by Eq. 3.15 with randomized step size α. As a rule of thumb, the algorithm has to balance all pixels in the neighborhood of pixels and converges in $c = 16\sigma^2$ steps, with $16\sigma^2$ being the neighborhood of pixels that have significant influence in the two-dimensional Gaussian kernel ($(2 \cdot 2\sigma)^2$ covering over 95% of the kernel).

Simulated annealing accepts (in the beginning) that some iterations produce worse results than the iteration before in order to avoid convergence to local optima in the optimization. This acceptance is reduced over time such that the algorithm converges "greedy" to the global optimum in the end. In our instantiation, we reduce the probability to accept worse solutions from 1.0 to 0.0 in $\frac{c}{2}$ steps (details can be found in (Kirkpatrick et al., 1983)). The best solution of all threads is then selected as final result. See Chapter 4 (p. 149) for a discussion of complexity and for an efficient algorithm.

Bounding constraints

Another issue are the limits of the display. sRGB does not support all the colors that are defined in perceptual color spaces such as DIN99. One solution is to integrate the borders of the defined sRGB colors as non-linear constraints in the optimization as presented by Lee et al. (2012). We select a heuristic for our iterative method, which samples the defined sRGB color space in DIN99. Pixels that become undefined in one iteration are assigned to the perceptually closest color. In the next iterations, the pixel and its surrounding pixels will adapt to this constraint.

3.4 Evaluation

There are several questions that need to be answered by quantitative user studies. First, does the perception model predict what the users perceive if they read color encoded data? Second, does the method presented in this chapter improve the accuracy of users reading and comparing color encoded data in real analysis tasks? And third, is this improvement significant? The following two experiments provide answers to these questions.

3.4.1 Experiment 1

The goal of the first experiment was to evaluate the perception model on the basis of pixel-cell based visualizations. We measured the accuracy of participants decoding colored information and compared these results to the predicted values of the perception model. Further, we measured whether the accuracy of participants will improve, if compensation is applied. Our hypotheses are:

H1 *Predicted values of the perception model are equal to participant results.* In the ideal case, the perception model decodes data values similar to a participant.

H2 *Participants are as accurate or better with our method than with the standard mapping.* Our method reduces the contrast effects; therefore, it improves the participant results.

Task

One-dimensional time series were visualized in a cell based visualization above and under an interactive drawing field (see Figure 3.11). The participants were

asked to estimate the quantity of each pixel cell and the trend of the time series (*elementary and synoptic identification*, see Section 1.6.2 (p. 27)). For this purpose, we provided an interface which allows to click pixels, to drag the line (trend) and to correct any errors by simply adjusting the line accordingly. The colormap was shown above the field for reference and participants were told that the lowest and highest values in the colormap correspond to the lowest and highest positions in the field.

Such complex tasks include biases. We see a gap between perceiving the data values on the screen and expressing the cognitive processed information on the drawing field. Also the participant may be influenced by the line drawn before. However, participants have to cognitively process sets of data values and recognize trends, which is realistic in real data analysis scenarios.

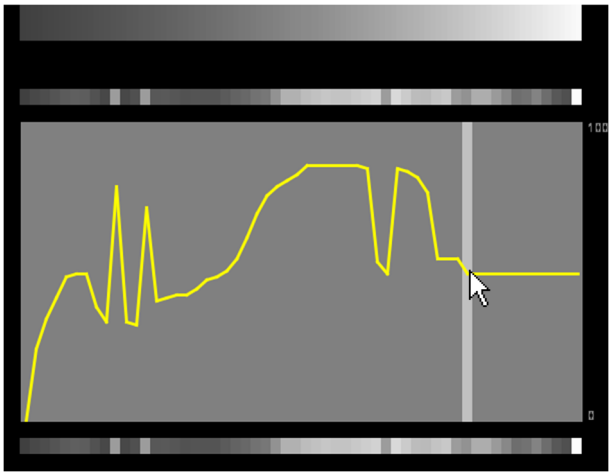


Figure 3.11: Image taken from [14]. *Task of experiment 1. Participants had to draw a line (shown in yellow) that estimates the value at each point in the time series shown above and below the drawing field. The colormap was given for reference at the top of the display.*

Experiment Factors

The experimental factors were colormap and color mapping (with or without our method). We selected one achromatic scale and three chromatic scales shown in Figure 3.12 on the basis of the guidelines in Chapter 2 (p. 31). All colormaps were perceptually equally spaced by interpolations in the DIN99 color

space. The achromatic scale continuously increased in lightness from dark gray to white. The first linear chromatic scale decreased in saturation from blue to white. The second linear scheme had equal lightness and only varied over hues from red to blue. The fourth *multihue* colormap was explicitly designed for *elementary identification and comparison* (see Section 1.6.2 (p. 27)) in order to measure if the compensation of contrast effects outperforms the state-of-the-art. This colormap varies over hues with continuously increasing intensity, and thereby, reduces the probability of contrast effects (Ware, 1988). Note, that some *multihue* colormaps such as the common *rainbow* colormaps share the problems of non-perceptual uniformity, false coloring and attention steering that are harmful in real analysis scenarios. These issues are reduced in the selected colormap (see Chapter 2 (p. 57)).

Experimental design

A user study with 40 participants was conducted. The experiment was split into two parts: In the first part, the data was collected to evaluate the perception model. In the second part, the mappings (with and without our compensation method) were alternated to measure how the results of participants change when compensation is applied.

First part

20 participants performed the task on different colormaps. The participants were randomly assigned to one colormap and trained on the tool and colormap. Each participant fulfilled two phases with a break in-between. In each phase, the participants performed two training tasks to ensure that they understood the colormap correctly followed by three real tasks.

Second part

The goal was to measure the improvement of accuracy of participants with our compensation method. Therefore, the experiment followed a within-subject

design and another 20 participants were randomly assigned to the colormaps. They were trained on the tool and then performed 2 phases with each 4 tasks alternating between the mappings in randomized order.

Participants

The participants (14 female, 26 male) were mixed graduated and non-graduated without background of information visualization but they were all familiar with infographics. The ages ranged from 19 to 57 with an average of 27.37 participants had normal or corrected to normal color vision. Three male participants were colorblind.

Data

The data originated from power consumption measurements of a Smart Grid environment, where the analysis of past data requires a faithful visualization of large volumes of time series data. A set of 40 similar time series was selected from this data source for our experiment. In each task, a time series was randomly selected from this set and visualized. The time series were normalized.

Apparatus

The study was performed under controlled lab conditions. The monitor was color calibrated with a resolution of 1920x1200 pixels. Each pixel cell had a width of $0.125°$ of the viewing angle. The ambient light was controlled to normal daylight conditions, without reflectance on the screen. The viewing distance was approximately 60 cm. The introduction, training and tasks were web-based and standardized.

Metrics

We measured the error of participants in the task (estimating the quantity of each cell). We, therefore, computed the Euclidean distance between the original time series and the participant results. In order to evaluate the perception model, the result values of the perception model were predicted for each time series-colormap combination. The difference of pairwise participant results, as well

as the difference of the perception model and participants were estimated by calculating the pairwise Euclidean distance. The time for a participant to finish each task was recorded.

Results

Figure 3.12 shows a summary of the results.

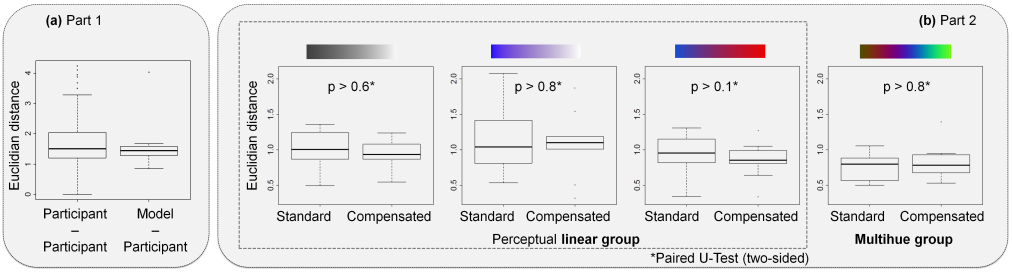


Figure 3.12: Image taken from [14]. *Results of experiment 1.* ***(a)*** *Pairwise distance measure for participant-participant and participant-model comparison.* ***(b)*** *Error of participants in the metric task using standard mapping and our compensation method.*

Evaluation of perception model

We found no significant difference (U-test: p>0.5) between the pairwise participant-participant distances (median=1.51, iqr=0.84) and the pairwise model-participant distances (median=1.44, iqr=0.30).

Evaluation of compensation method

In our experiment, there was no significant difference (H-Test: $p > 0.6$) between the linear (intensity, saturation and hue) colormap-groups. The error of the linear-group (combined: median=0.98, iqr=0.34) was significantly greater (U-Test: $p < 0.01$) than the error of the multihue-group (median=0.80, iqr=0.24). The results of the within-subject comparisons (see Figure 3.12b)) revealed that there was no significant shift of error location and ratio of error scales between our method and a standard mapping. However, the inter-quartile ranges and medians were smaller when compensation was applied for all groups.

Efficiency

The choice of colormap had significant influence on the efficiency of participants performing the task (H-Test: $p < 0.001$). The red-blue colormap group performed worse with a median of 209 seconds (iqr=84.75). The participants mentioned that they had problems to interpret differences in the middle of the colormap. The multihue group (median=94, iqr=33) was slower than the other linear groups (combined: median=81, iqr=56), however, there was no significant difference (U-Test: $p > 0.2$).

3.4.2 Experiment 2

The goal of our 2nd experiment was to measure the contrast effects on the estimation and comparison of metric quantities, and how participants improve when our method is applied. Our hypotheses were as follows:

H3 *Participants assign more data points to the correct data ranges with our method than with standard mapping.* Using the method, we expect participants not to overestimate and underestimate values in high frequency areas of the image.

H4 *Participants increase the number of correct comparisons of data points with our method in comparison to the standard mapping.* We assume that participants perceive the differences of data points correctly with our method.

Tasks

The task was inspired by Ware's metric task (Ware, 1988) for colored data visualizations. Ware showed the participants a visualization and asked participants to match the color at a given pixel (indicated with a cross-hair) with a given set of colors. We adapted this method to our experiment but extended it by elementary comparison in the intentional presence of contrast effects (this task,

thereby, measures the error of users in the *elementary identification and comparison tasks*, see Section 1.6.2 (p. 27)). According to Ware (1988), participants make up to a 20 % error in the estimation of the correct quantity if a linear colormap is used. Since we were measuring cases of extreme contrast effects, fine granular data ranges would not increase the expressiveness of our measurements. Therefore, the data range was split into three equally sized parts (low, middle, high). Two data points were marked that shared the same data value but had different surrounds (see Figure 3.2). We ensured that the data values were at least 5 % away from the data range border. Users were asked to assign the points to the data ranges (elementary identification). If a participant assigned the points correctly to the same data range, we asked which of the data points was higher or whether they were equal (elementary comparison).

Experimental design & metrics

The experiment was a within-subject design with 20 participants of the first experiment (part two). After finishing experiment 1, each participant was shown 32 different visualizations, as illustrated in Figure 3.2 for the time series data (encoded with their colormap of experiment 1). Our method was applied in half of the visualizations with randomized order of the mappings. For each of the 32 tasks we counted the number of correct data value assignments and comparisons.

Results

The summary of results is illustrated in Figure 3.13.

Elementary Identification (reading metric quantity)

As in experiment 1, there was no difference between the linear colormap groups (H-Test: $p > 0.3$). Again, the linear groups (combined: median=0.81, iqr=0.44) assigned significantly less (U-test: $p < 0.001$) data points to their correct value range than the multihue group (median=1.0, iqr=0). The within-subject compar-

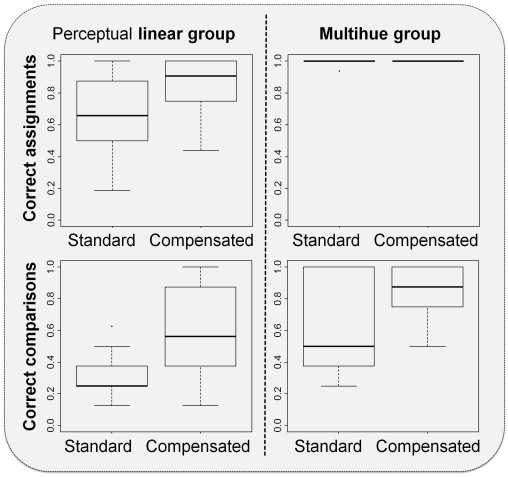


Figure 3.13: Image taken from [14]. *Results of experiment 2.* ***(Top)*** *The percentage of correct data value assignments increased when our method was applied on perceptual linear colormaps.* ***(Bottom)*** *The percentage of correct data value comparisons did increase with our method.*

isons revealed a significant increase in the number of correct participant answers (paired U-Test: $p < 0.001$) with our method (median=0.91, iqr=0.25) compared to standard mapping (median=0.66, iqr=0.38) for the linear groups.

Elementary Comparison

Users of all groups showed a significant increase (paired U-Test: $p < 0.05$) of correct answers with our method. In the linear groups the median percentage of correct answers increased from 0.25 (iqr=0.125) to 0.56 (iqr=0.5) and in the mutlihue group the median increased from 0.5 (iqr=0.5) to 0.875 (iqr=0.22).

3.4.3 Discussion

We found that the predicted values of the perception model were close to the participant results. Uncertainties of the complex task and abilities to master the colormap and tool varied from participant to participant, which was expressed in the pairwise differences. Since there was no significant difference between

the participant-participant and participant-model distances, we conclude that the model behaves like an average participant and thus, we can approve H1.

Our study confirms the guidelines in Chapter 2 that colormaps that vary over hues with linear increasing intensity perform well for the elementary identification and comparison of color encoded metric quantities. In all groups, the inter-quartile ranges and medians decreased with our method, which indicates that the accuracy of all participants was improved and thus, H2 can be approved. Even in combination with our method, grayscale mapping was not as accurate as the standard multihue colormap, which indicates that there might be a maximum of accuracy that can be achieved by each colormap. This confirms the guidelines of the community that colormaps must match the visualization task and data, even with our compensation method.

The second experiment revealed a significant influence of contrast effects on linear scales as already discussed and quantified by Ware (1988). The participants were able to identify in only 66% of the trials the correct value range, which was increased to 91% with our method. In contrast, participants were correct with the multihue colormap, because the granularity of data ranges was too coarse for this colormap, which was explicitly designed for identification and comparison of color encoded data values. In the task of comparing data values under the influence of contrast effects, the accuracy of users increased from 25% to 56% for the linear colormaps. Even with the well-designed multihue colormap, most of the participants were only correct in 50% of the trials in the comparison task, which was increased to 88% with our method. Thus, we can conclude that our method significantly improves the accuracy of participants in both tasks, which confirms H3 and H4.

Another issue is raised by the efficiency of participants. The multihue colormap did outperform the other mappings in terms of accuracy in the metric task and the low probability of contrast effects. However, participants were 14 % faster with the grayscale and saturation based colormaps than with the multihue colormap. This might be due to the issues of pre-attentiveness, since it is pre-attentive to identify trends in perceptual linear colormaps (Section 2.3.2 (p. 49)).

3.5 Applications

3.5.1 Purple America Map

The Purple America map of Vanderbei (2012) visualizes the results of the 2012 presidential election in the United States. Figure 3.14 shows a detail of the western part of this map. The visualization uses color to show the proportion of votes for Democrats (blue), Republicans (red), and others (green) for each congressional district. The color of a district results from a mixture of these colors corresponding to the percentage of votes. This color choice is solely based on the colors associated with the parties and does not consider perceptual effects.

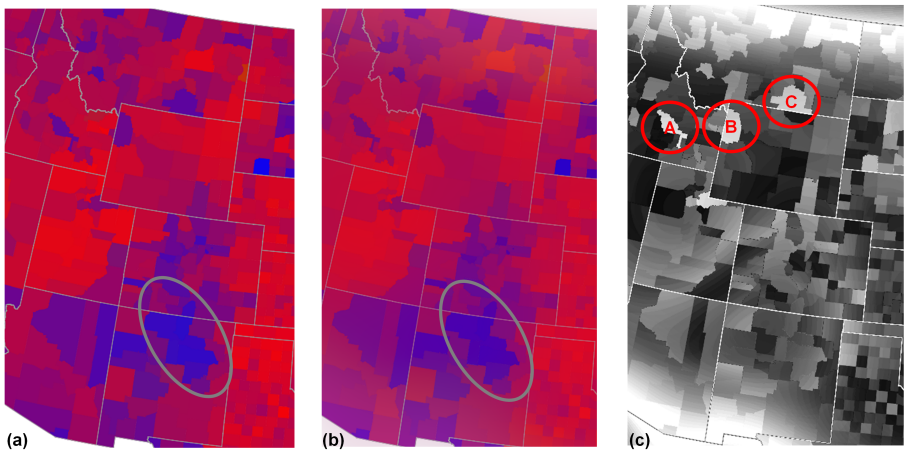


Figure 3.14: Image taken from [14]. *Detail of Vanderbei's Purple America map of the 2012 presidential election in the US (Vanderbei, 2012). The images show the original **(a)** and compensated **(b)** visualization. The normalized difference between (a) and (b) is shown in (c). The labels in c) mark the districts or counties Baine/ID (A), Teton/WY (B), and Big Horn/MT (C). The colors in (a) appear crisp and saturated due to color contrast effects. Our compensation removes color contrasts and the resulting image (b) appears less saturated with dull colors but represents the data faithfully. In (b) the share of blue increases from (B) to (A) and (C), which accords to the data.*

This usage of blue and red leads to very strong color contrasts, which gives rise to small areas of one color appearing stronger when surrounded by the other color. For instance, the color of the districts Baine/ID (A) and Big Horn/MT (C) appear similar in Figure 3.14(a). Looking into the numbers, the distance between (A) and (C) is 5.7 %. However, the distance of (A) to (B) is with 4.3 % quite similar but (A) and (B) appear different. The contrast between the red surround and the value of (A) causes the district to appear more bluish and thus, similar to (C).

The result of our compensation method on this image is shown in Figure 3.14(b). The compensation changed the color of almost all districts. The magnitudes of differences are shown in Figure 3.14(c). In total, the compensation method reduced the contrasts in the image. Not only the contrasts between districts are reduced but also contrasts between the border of the map and the white background. As an effect of the compensation the Purple America map appears even more purple than before. The real differences of the colors of the districts (A), (B), and (C) are more accurately represented. It is clear that the share of blue increases from (B), to (A) and (C), which is a more faithful representation of the data.

3.5.2 News Visualization

Another example of compensating contrast effects can be found in Figure 3.15, which shows a news visualization of Wanner et al. (2009). The original visualization in Figure 3.15(a) plots a semi-transparent triangle for each news item. The color of the triangle (red, white, blue) is mapped to the sentiment of the item (negative, neutral, positive). The different rows represent different news items at different days. The technique clusters news visually. When several items of news of the same category are published at the same time, the triangles overlap and a continuous block of news becomes visible.

In Figure 3.15(a), we can see that the differing contrast between the triangles and their background alters the appearance of triangles. The perceived satura-

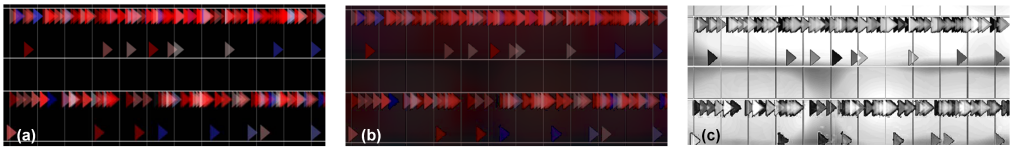


Figure 3.15: Image taken from [14]. *Detail of a news visualization in (Wanner et al., 2009) using colored triangles with low alpha value for representing single news items in time. The images show the original a) and compensated b) visualization. The normalized difference between a) and b) is shown in c). Depending on the color, single news items in a) appear as bright as dense episodes of news. Our compensation in b) compensates the contrast effects and corrects the impression of single news items.*

tion of single triangles depends on their color and their neighbors. For instance, the triangles in the fourth row appear to have different saturations. Some appear as saturated as the groups of triangles in the first or third row, which is actually a perception error. Our compensation method takes the color contrasts into account and generates Figure 3.15(b). In this image, the triangles in the fourth row appear to have the same saturation as in the areas with many triangles (in the first and third row). In the difference image in Figure 3.15(c), one can see that the compensation method has changed the triangles differently to compensate for the contrast effects. Through the black background, we overestimate the brightness and differences of triangles in the original image, which is reduced by our method.

3.6 Discussion & Future Work

In this chapter, we present the first methodology and method compensating harmful physiological bias such as contrast effects based on color appearance models and optimization algorithms. We present the necessary cost functions and heuristics to reach the optimization goal. Thereby, the method satisfies the requirement of *faithfulness* for effective color encodings. Together with the guidelines of Chapter 2, effective color encodings can be provided.

Our experiments show that with our method users double their accuracy in *comparison* tasks and significantly improve their accuracy in *identifying* data values encoded by color. However, there are tasks in which compensation of contrast effects is not desired. Further, we see many open research questions since this method is the first step for compensating physiological biases in information visualization.

When to use the technique?

Our studies show that our method significantly increases the accuracy of users reading and comparing data values. Therefore, it should be integrated in every data visualization that uses color to encode metric quantities, which should be accurately perceived by the user (especially in elementary or synoptic identification and comparison, see Section 1.6.2 (p. 27)).

For localization tasks, the aim is to guide the users attention towards important data patterns. This is typically well supported by increasing the contrast of important data points to their surround. This shows that this task exploits contrast effects to reveal important information. However, the colormap design cannot prevent the occurrence of "not intended" contrast effects, which lead to a false highlighting due to the surround. Such effects can only be compensated within the rendered image with our method for compensating contrast effect. Beside these harmful contrast effect, which should be compensated, contrast effects can also be applied on purpose. This relates to *visual boosting* in Chapter 5, in which contrast effects are used to reveal information that is invisible.

The eye naturally amplifies contrast to detect edges in a scene, which supports the identification of natural forms and shapes (Ware, 1988). Therefore, the task to detect natural forms and shapes in visualizations is an example where compensating contrast effect would reduce the effectiveness of our eye.

Local-adaptive Perception Models

We have experienced some limitations of the iCAM perception model of Fairchild and Johnson (2004). Since our method behaves like a contrast low pass filter and uses a distinct kernel size for the whole image, we often have the impression that

the final results of the method are not as clear as the original data visualizations. High-frequency achromatic structures sometimes show artifacts in these cases, because the method induces color. Therefore, we hypothesize that the perception model must adapt to local spatial frequencies. From a computer science perspective it would make sense to compute the spatial frequency of the image at each pixel and to locally adapt the kernel sizes (see Section 3.3.2 (p. 120)). However, it is not clear if this is perceptually sound for predicting contrast effects. There is evidence that contrast sensitivity decreases with higher spatial frequency (Watson and Ahumada Jr, 2005), but there is also evidence that the impact of contrast effects increase for smaller (high frequency) stimuli (Shi et al., 2013). Thus, it remains an open research challenge to model the perception of contrast effects for different spatial frequencies. However, the perception model is an exchangeable component of our methodology, thus, we can easily adapt our method to an advanced perception model in the future.

Pattern Preserving Metrics

It is our goal to provide an application independent method, which can be used for post-processing the rendered visualization to provide a faithful data representation. Therefore, our cost functions are image-based metrics that preserve the accurate perception of the pixel colors and neighborhoods of pixels. Thereby, they also preserve the accurate perception of single data objects even if these visual objects consist of multiple pixels.

Our method assumes that the color encoding already *faithfully* captures the characteristics of the data and aims to optimize the final rendering such that the color encoding is *faithfully* perceived. However, at the cost of being application independent, our metrics cannot preserve patterns that are beyond the color encoding or input image. Thus, the method cannot preserve data relations between objects that are found in different spatial locations in the image, e.g., preserve the data difference of two (spatial distant) objects, preserve the average color of cluster items, or preserve the visual distinction of all classes within the data. It would be interesting to integrate such application dependent metrics (as presented in Section 2.4.2 (p. 81)) into our method of contrast compensation.

However, this would conflict with our goal to provide a general application independent method that can be applied as post-processing method to any visualization Further, we experienced that the resulting colors of the method are accurately perceived but appear dull and not aesthetic. Therefore, we see a challenge in preserving perceptual accuracy but also aesthetic colors. It would be interesting to integrate metrics that preserve the vividness of colors or even recolor the whole visualization according aesthetic and accuracy goals.

Color Blindness

The iCAM perception model of Fairchild and Johnson (2004) aims to provide an average model for normal color vision. It operates on the lowest level of color perception (how the different cones types in the human eye process the amount of incoming light). However, it is not clear if this correctly models perception contrast effects for color vision deficiency (CVD). There are different types of color blindness, e.g., prot-, deuter-, and tritanopia. They are due to genetic defects causing that one type of cone is missing. Other forms of CVD are anomalous CVD, which shift the sensitivity of photo-receptors into different from normal wavelengths of light; thus, to another spectrum.

In our experiments participated three male persons with CVD. Interestingly, there was no significant difference between their results and the results of other persons. Their accuracy also improved when compensation of contrast effects applied, which implies that our method is able to compensate contrast effects for CVD. However, the sample size is not representative. Further, it was not recorded, which type of CVD was present (either deuteranopia or deuteranomalous). Therefore, it remains unanswered if the perception model is able to predict the perception of contrast effects for persons with CVD. And if no (which is very likely), there remains the open research challenge to provide contrast effect compensation for color blind persons.

Other open questions: How efficient is our compensation method? How to capture individual characteristics of color and contrast perception? How to accommodate for different environmental settings such as ambient light and viewing distance? These questions are answered in the next chapter.

4

Personalized Contrast Effect Compensation

Contents

This chapter is based on the following publications and major parts of the sections also appeared in the following publications:

[11] Sebastian Mittelstädt and Daniel A. Keim. *Efficient Contrast Effect Compensation with Personalized Perception Models*. Computer Graphics Forum, 34(3):211–220, 2015.

[14] Sebastian Mittelstädt, Andreas Stoffel, and Daniel A. Keim. *Methods for Compensating Contrast Effects in Information Visualization*. Computer Graphics Forum, 33(3):231–240, 2014.

For the division of responsibilities and work, as well as a statement of contributions in these publications, please refer to Section 1.5 (p. 18).

The following contributions go beyond the published work:

1. Integration of contrast sensitivity into the perception model, which models the view distance of observers.
2. Display dependent perception models for adapting contrast effect compensation to different environment settings.
3. Methods to boost high-frequency information to account for contrast sensitivity.
4. Methods to compensate contrast effects depending on the view distance.

Chapter Abtract & Structure

A method that compensates for contrast effects has been introduced in the last chapter, which significantly improves the users' accuracy in reading and comparing color encoded data. The method utilizes established perception models to compensate for contrast effects, assuming an average human observer and ideal environment settings (color calibrated monitor, standardized ambient light "D65", and an optimal viewing distance of "60cm"). However, it is clear that ideal environment settings are not realistic in common offices or special hardware settings. Further, the perception and cognition is different from individual to individual.

The main research question raised in this chapter is "Is the perception of contrast effects different for each individual and if yes, how can we accommodate for this?" In order to measure the perception of an individual, the methods must provide interaction. However, the instantiation of the methodology in Chapter 3 is designed to find the optimal solution (high effectiveness) and has a runtime of several minutes. This is higher than the latency limit of interactions. Liu and Heer (2014) found that users are not willing to accept a higher latency than 500ms. Therefore, there is a need for algorithms that perform contrast effect compensation within this limit. These algorithms would enable interactive visualizations and experiments such as the personalization of contrast effect compensation.

In this chapter, we introduce the need for interactive contrast effect compensation in Section 4.1. Related approaches to personalize perception models are discussed in Section 4.2. We propose an efficient algorithm in Section 4.3 to overcome the major limitation of our original method, which is a runtime of several minutes. With the use of efficient optimization and surrogate models, we are able to reduce runtime to milliseconds, making the method applicable in interactive visualizations and experiments. We provide experiments that show a significant difference in the perception of individual users and introduce methods to personalize contrast effect compensation in Section 4.4. Further, we show that personalization outperforms our original method with a user study in Sec-

tion 4.5. Section 4.6 provides means to parameterize the method to the target application and further, presents a use case in which the method was deployed. Section 4.7 provides further discussions and future work.

4.1 The Need for Efficient Compensation Algorithms

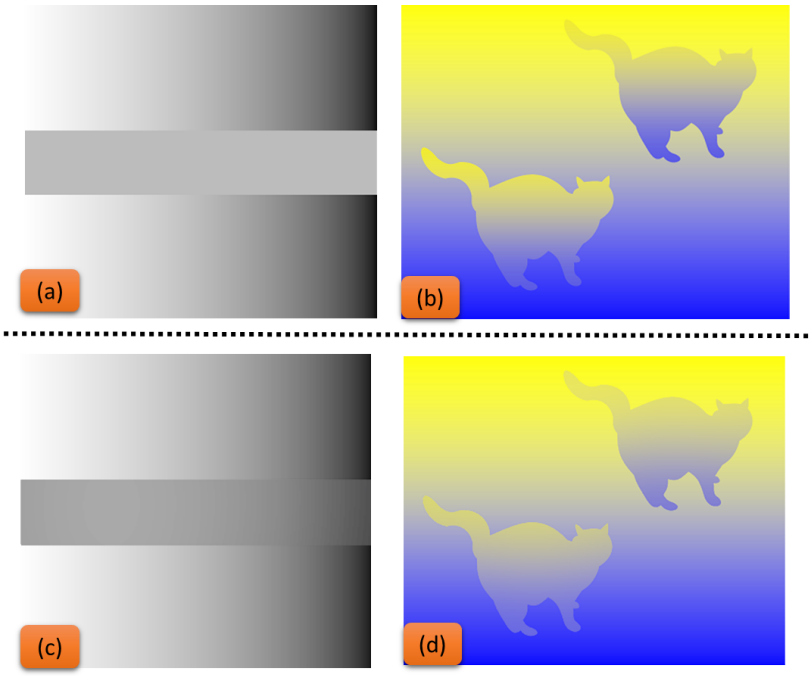


Figure 4.1: Image taken from [11]. *Examples of contrast effects. (a) The ends of the gray bar and also the cats (b) are perceived differently, even though they are equal. (c) and (d) show compensated results of the "original method" of Chapter 3.*

The last chapter presented our pixel-based optimization method that is application independent and can be used as a post-processing method to compensate for contrast effects in images or visualizations. The bias of contrast effects was

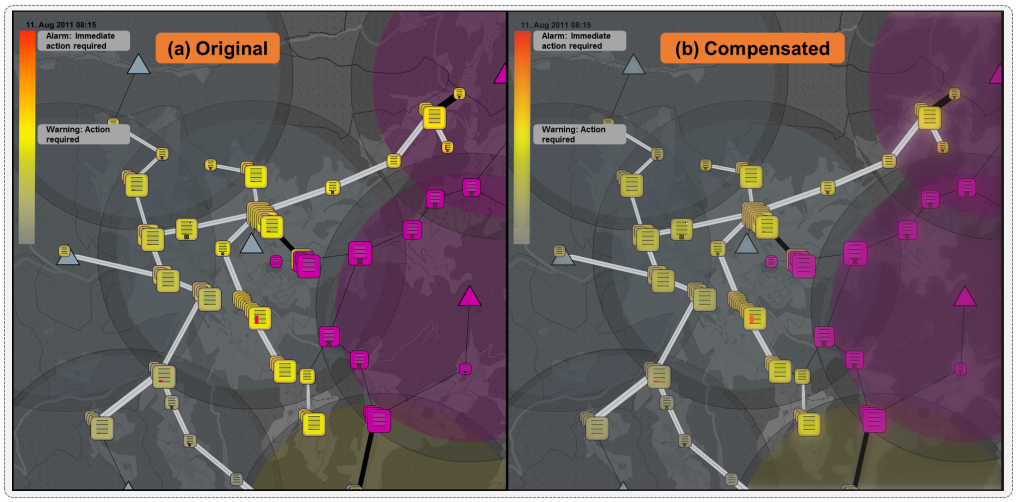


Figure 4.2: Image taken from [11]. *Overview of a smart grid [16]. Transformer stations (rectangles) are connected via power lines and linked to the communication infrastructure (triangles). While gray indicates normal operation mode, yellow elements on the screen reveal a severe situation (violet stations are destroyed due to a debris avalanche in the eastern area). Due to contrast effects, the elements in (a) seem to be more critical than they actually are, causing that operators unnecessarily try to reconfigure the system, which increases operational costs. In (b), contrast effects are compensated showing (accurately) a less severe situation.*

also experienced in an interactive control room scenario for monitoring multiple critical infrastructures with high resolution displays [16] (see Fig. 4.2). In this application, color encodes the status of infrastructure elements and reveals to crisis managers when, where and how to act. Thus, the color of elements must be accurately perceived. The method proposed in Chapter 3 to compensate for contrast effects takes four minutes to compute the final result in this application, which is inappropriate for interactive visualizations.

In this chapter, we propose an efficient algorithm that performs contrast effect compensation in almost real time. We overcome this limitation by introducing an efficient algorithm that uses efficient optimization and surrogate models. Thereby, the reader can learn how surrogate models can be applied to reduce the runtime of complex image algorithms and optimization problems.

Liu and Heer (2014) found that analysts perform significantly worse in analysis tasks if interactions are delayed by 500ms. Our novel algorithm renders the

high resolution images of the sample application in 360ms, which is below the latency limit of 500ms for interactions. This algorithm enables, thereby, interactive visualizations and experiments. Thus, it also enables the personalization of contrast effect compensation.

Another limitation of our original method is that the applied perception model assumes optimal conditions such as D65 ambient light, a color calibrated monitor and an "average" user. However, there is evidence that the perception of contrast effects is different from individual to individual (Ekroll and Faul, 2009). Interestingly, they showed that a simple model based on *von Kries adaptation* (von Kries, 1905) can explain these differences. Therefore, we introduce a novel perception model based on the iCAM framework (Fairchild and Johnson, 2004) and *von Kries adaptation* that can be personalized to the individual user and adapted to external influences (ambient light and viewing distance) by interaction. We are able to prove that the personalization of contrast effect compensation significantly increases the accuracy of users by 29% in reading color encoded information in comparison to the standard technique.

4.1.1 Contributions

We claim the following contributions:

1. An efficient algorithm to compensate for contrast effects.
2. Methods to personalize contrast effect compensation.
3. Integration of contrast sensitivity into the perception model, which models the view distance of observers.
4. Display dependent perception models for adapting contrast effect compensation to different environment settings.
5. Methods to boost high-frequency information based on contrast sensitivity.

6. Methods to compensate contrast effects dependent on the view distance.

7. An evaluation of personalized contrast effect compensation with a user study.

4.2 Related Work

Kindlmann et al. (2002) proposed a method to adapt luminance of colormaps to different output devices, different environments, and individual color perception. Thereby, luminance variation of colormaps were personalized by a method that exploits our cognitive processes to detect human faces (Rogowitz and Kalvin, 2001). They could show that there are significant individual difference in color vision and are able to prove the soundness of their approach.

Flatla and Gutwin (2011) proposed an efficient in-situ calibration for personalizing color-vision-deficiency (CVD) models (situation-specific models). These models measure individual characteristics of color differentiation in any environment and are able to capture congenital CVD, situation-induced CVD, and acquired (permanent or temporary) CVD. Based on these personalized models, they are able to provide effective recoloring methods that optimizes the coloring (with a focus on providing well-differentiable/categorical colors) of visualizations (Flatla and Gutwin, 2012) and web-pages (Flatla et al., 2013) for an individual observer. They found that participants match colors significantly faster with their personalized method than with any other recoloring method.

The specification of MPEG 21 DIA discusses the personalization of multimedia content based on individual preferences for content display/rendering, quality of service, and configuration and conversion with regard to multimedia modalities (Vetro, 2004). Nam et al. (2005) integrated personalization of color vision deficiency, low vision impairment, and color temperature in the MPEG 21 framework.

Bennett and Quigley (2011) build an model of individual human eyesight (the virtual eye of a user) which is able to capture individual differences with various test charts (predominantly in contrast- and spatial sensitivity). They propose metrics to estimate differences between the original and perceived design and can predict what parts of a visual design are easy or difficult to see. The authors present how, for example, font sizes can be increased to adapt visualization designs according to the virtual eye of a user.

Itoh and Klinker (2015) enhance visual acuity by optical head mounted displays. They provide a method for defocus correction by presenting a compensated image, e.g., in which contrasts are enhanced for readability inside the optical device.

Green and Fisher (2010) provide measure for locus of control, extraversion, and neuroticism in visual analytics systems. They show that these measures can predict completion times of visual analysis tasks. Further, they found that personality measures also predicted the number of insights participants reported and highlight that these measures should be used to personalize interaction design in visual analytics.

Toker et al. and Streichen et al. (Toker et al. (2013), Steichen et al. (2013), Steichen et al. (2014)) show that eye-tracking can be used to predict the analysis task of users in information visualization. Therefore, they perform eye gaze sequence analysis and provide evidence that it is possible to predict the users analysis task with eye-gaze measurement on bar and radar graph visualizations. This reveals the great potential of integrating perception models combined with eye-tracking in order to support users in their current analysis task by adapting the whole visualization system. However, the realization of such a system remains one of the biggest research challenges in visual analytics.

4.3 Efficient Compensation of Contrast Effects

A major limitation of our compensation method in Chapter 3 (we will refer to this method as the "original method") is a runtime of several minutes. This section introduces a novel method that uses efficient optimization and surrogate models to reduce complexity and runtime to milliseconds.

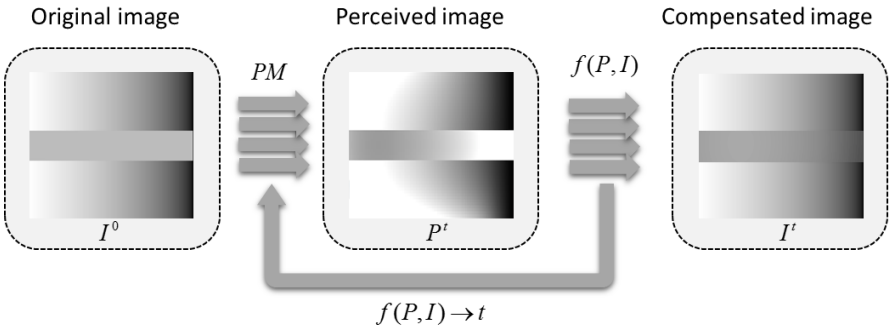


Figure 4.3: Image taken from [11]. *The algorithm computes the perception model PM, the bias with cost function f, and iteratively reduces the bias in an optimization process. Our algorithm parallelizes PM and f and also introduces a convergence threshold t.*

4.3.1 Why "good" solutions are "good enough"

Our original method applied *line search* optimization that uses the gradient ΔG (Section 3.3.3 (p. 124)) to reduce contrast effects in each iteration. It is combined with *simulated annealing* in order to avoid convergence to a local minimum of the cost functions. The cost functions are based on estimating the color differences between the perceived and original image (see Eq. 3.3 (p. 115)). The method tries to converge the cost functions to zero.

However, since there is a limit for humans to perceive color differences (just-noticeable-difference: *JND*) (MacAdam, 1942), convergence to zero is not obligatory to provide a "good" solution. This indicates that local minima may be as good as global minima since humans cannot perceive any difference between the solutions. This allows a shortcut for the optimization algorithm to converge the cost functions to a threshold. The selection of the convergence threshold is dependent on how the different cost functions are handled by the selected optimization algorithm. We use *golden section line search* (Press et al., 2007), which converges to local minima. As in the original method, we aggregate the cost functions to an equally weighted sum. The threshold of convergence t is selected with $t = 1_{\Delta E}$ in DIN99 (see Eq. 3.2 (p. 114)) where $1_{\Delta E}$ approximates the JND for simplicity (see Mahy et al. (1994) for accurate experiments).

The complexity of our algorithm can be estimated for each part (Figure 4.4). The perception model requires a convolution of the image, which has a complexity of $\mathcal{O}(nk^2)$, with n being the number of image pixels and k being the size of the kernel in one dimension (see below). The estimation of the gradient does also require a second convolution with $\mathcal{O}(nk^2)$. The cost functions are of $\mathcal{O}(n)$ linear complexity. As a rule of thumb, the *golden section* algorithm converges in approximately $c \approx k$ steps. Thus, the algorithm has $\mathcal{O}(c(nk^2 + nk^2 + n)) \approx \mathcal{O}(nk^3)$ complexity.

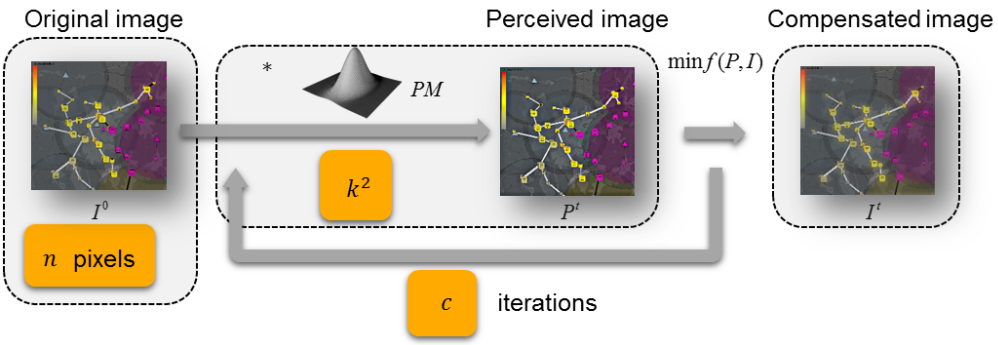


Figure 4.4: *Complexity of the algorithm.*

Convolution

One of the most expensive parts of the algorithm is the computation of the perception model and the estimation of the gradient. Both involve convolution of the original image I.

Recap of the perception model in Section 3.3.2 (p. 118): The image is separated into the LMS channels (red, green, blue light). Each channel of the image is convolved with a Gaussian filter K to model the perceived surround S. K depends on the spatial frequency of the image; thus, depends on the application. We propose in Section 3.3.2 (p. 120) to filter the image with several Difference-of-Gaussians (DOG) kernels. K is then set to the kernel of the DOG that determines the most dominant (average) spatial frequencies of the image.

There exists a variety of efficient convolution algorithms such as convolution in the frequency domain, recursive-, or separable convolution. Transformation to the frequency domain is of $\mathcal{O}(nlog(n))$ complexity and has to be repeated in each iteration. Recursive convolution is computationally the fastest version. However, this method lacks in accuracy. The third option is separable convolution. Gaussian kernels are separable such that a two-dimensional kernels can be expressed by two one-dimensional kernels. This reduces the complexity from $\mathcal{O}(nk^2)$ to $\mathcal{O}(nk + nk) \approx \mathcal{O}(nk)$. Since the perception model requires an accurate convolution and the typical kernel size is small (compared to the transformation into the frequency domain), we decided to use separable convolution. This reduces the complexity of the algorithm from $\mathcal{O}(c(nk^2 + nk^2 + n))$ to $\mathcal{O}(c(nk + nk + n)) \approx \mathcal{O}(nk^2)$.

Massive Parallelism. The perception model, the cost functions, and the estimation of the gradient can be performed on each individual pixel independently. This allows transferring the computation to the GPU. The runtime is, thereby, significantly reduced by the degree of parallelization. We implemented the method in OpenCL (Stone et al., 2010) in order to minimize hardware dependencies.

4.3.2 Compensation with Surrogate Models

Surrogate models (Sedlmair et al., 2014) approximate the result of a computational model with reduced complexity or calculation on a subset of the input data. Our surrogate model reduces the complexity by sampling the input image. The main idea is to learn from a sampled image how to compensate for contrast effects and then to apply this to the original image.

Sampling

Sampling is a well known method to reduce the complexity of signals and images. If the *sampling theorem* (sampling rate $f > 2f_{max}$ with f_{max} being the maximum frequency in the original signal) is satisfied, the original signal can be reconstructed without information loss. Typically, f_{max} is very high. This does not allow compressing the image to major extend since f would need to be high as well. However, by violating the sampling theorem, reconstruction would result in artifacts (see Fig. 4.5(d)). Therefore, the common approach in image sampling is to limit f_{max} and to filter high frequencies (higher than f_{max}) with a low pass filter (Turkowski, 1990).

Likewise in image scaling, pixels are interpolated for up or downscaling (Kopf and Lischinski, 2011). However, low pass filtering or any other interpolation of the input image is harmful for the compensation algorithm because interpolation between pixels also modifies contrasts without considering human perception. Thus, the input for our algorithm would be harmed, resulting in artifacts. However, sampling without limiting f_{max} typically leads to a violation of the sampling theorem and thus, also leads to artifacts in the sampled signal (high frequency noise that was sampled) and artifacts in reconstruction (high frequency information that was not sampled). High frequencies in the sampled image (artifacts) are smoothed in the surround calculations of the perception model and

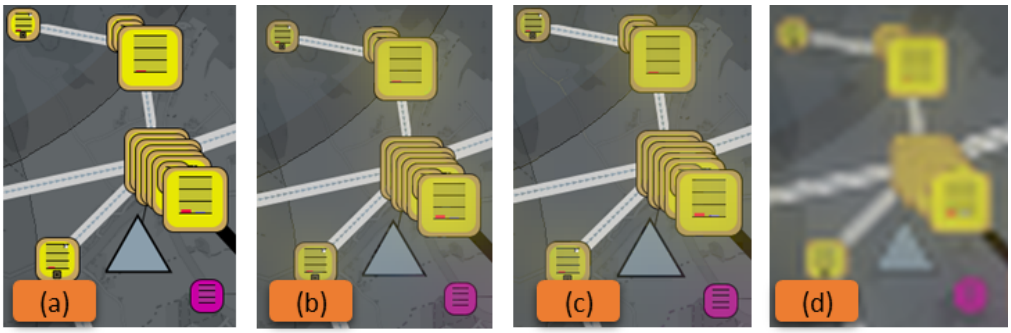


Figure 4.5: Image taken from [11]. *(a) Original image. (b) shows the compensated image without sampling in comparison to reconstruction with (c) our method and (d) the Lanczos filter method (Turkowski, 1990) (both with 7px sampling interval).*

thus, do not critically influence other pixels. However, the reconstruction of the final image requires methods to handle aliasing effects, which is discussed in the next section.

The additional costs of sampling is a computation of $\mathcal{O}(n/M)$ complexity with M being the sampling interval. However, sampling reduces the complexity of the overall algorithm to a major extend (Figure 4.6). First, the number of pixels is reduced quadratically to n/M^2. Second, the size of the kernel of the perception model is significantly reduced to k/M. Since k/M is typically very small (≈ 10), the influence of the convolution can be omitted in the complexity considerations. This reduces the complexity from $\mathcal{O}(nk^2)$ to $\mathcal{O}(nk/M^2)$.

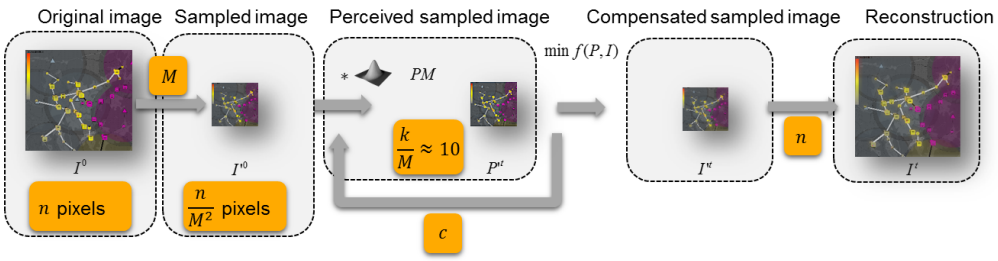


Figure 4.6: *This figure shows how sampling reduces the complexity quadratically.*

Reconstruction

As stated before, the reconstruction step is critical since aliasing effects must be handled. The compensation algorithm is applied on the sampled image I_s. The result is a sampled image I_s^C, in which contrast effects are compensated. The challenge is to compute the compensated version I^C of the original image I. This cannot be reconstructed from the compensated sampled image I_s^C with standard methods since the sampling theorem is violated. Therefore, any interpolation such as scaling or reconstruction by low pass filtering would result in artifacts (see Fig. 4.5(d)).
To reconstruct the image we state the following assumptions:

A1 Pixels at similar locations share similar surrounds.

A2 Pixels affected by the same contrast effect at similar locations can be compensated in the same way.

The idea of our reconstruction is that the algorithm learns how to compensate for contrast effects based on the sampled image and then applies the compensation to pixels in the original image by utilizing A2 (Fig. 4.7). Therefore, in the reconstruction phase the algorithm has first to apply the perception model on the original image I in order to estimate the perceived image P. The perception model requires to model the surround S for each original pixel, which is expensive. However, by following A1 we can assume that the nearest neighbor $I_{s,p}$ in the sampled image of each original pixel I_p has the same surround. This reduces the calculation of the surround to a simple lookup $S_p = S_{s,p}$ and requires no additional efforts since the compensation algorithm has already computed $S_{s,p}$ in the last iteration. Once the surround for each pixel is known, the algorithm can directly estimate the perceived image P and thus, the direction of contrast effects with $\Delta F = P - I$.

Due to A2, we can compensate pixels of the original image with pixels of the sampled image that are effected by the same contrast effect and share similar

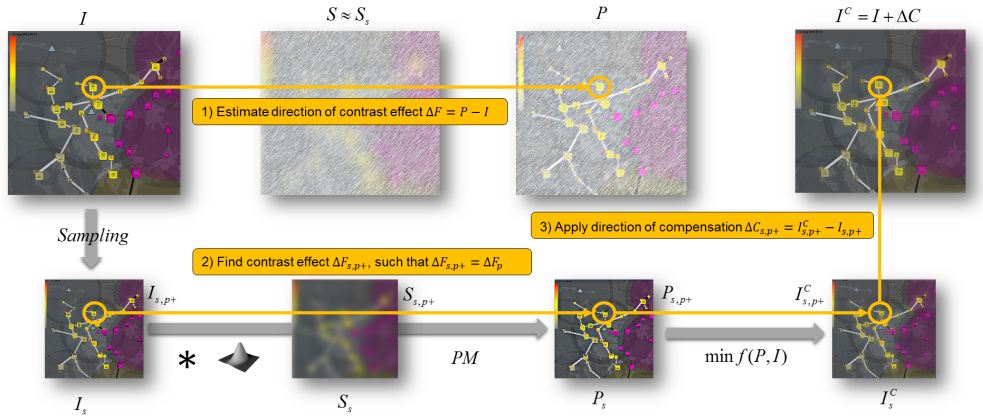


Figure 4.7: Image taken from [11]. *Contrast compensation with surrogate models. The method applies contrast compensation on the sampled image and "learns" how to compensate for contrast effects. The method determines the contrast effects for each original pixel, finds an equivalent sampled pixel in the (sampled) proximity, and applies the compensation. The algorithm adjusts the parameters automatically to provide an accurate solution without artifacts.*

locations. The algorithm starts the search for an equivalent pixel $I_{s,p+}$ at the nearest neighbor in the sampled image and continues the search in a spiral pattern. The algorithm stops, if the directions of contrast effects are similar (e.g., estimated by the cosine similarity: $cos(\Delta F_p, \Delta F_{s,p+}) > 0.99$). According to A2, we assume that pixel I_p will be compensated in the same way as $I_{s,p+}$ and thus, apply the direction of compensation $\Delta C_{s,p+}$ to the original pixel with $\Delta C_s = I^C_s - I_s$. The "extent" of compensation is scaled by the "extent" of the contrast effects (Eq. 4.1). This allows estimating the compensated image with $I^C = I + \Delta C$.

$$\Delta C_p := \frac{|\Delta F_p|}{|\Delta F_{s,p+}|} \cdot \Delta C_{s,p+} \tag{4.1}$$

This approach reduces the amount of artifacts significantly since no interpolation is applied that would lead to aliasing effects. High frequencies of the original image, that are missing in the sampled image, can also be compensated if the algorithm finds an equivalent pixel affected by the same contrast

effects (even if the pixels share not the same color).

The adaption of one pixel can have significant impact on the perception of other pixels. Therefore, this reconstruction has a limit in the spatial proximity requirement of A2. If this is violated (the algorithm finds no equivalent pixel in the proximity but far away), then the compensation of this pixel does not correspond to the compensation of the pixels in its surround, which results in artifacts. To overcome this issue, the search of the algorithm can be stopped if no pixel is found in, for example, the 25-neighborhood of a pixel in the sampled image. The compensation could then be approximated by state-of-the-art reconstruction, for example, a Lanczos filter (Turkowski, 1990). However, in our experiments, we found that this gives poorer results than an unlimited search by our reconstruction algorithm (see Fig. 4.5). The quality of reconstruction can be increased by increasing the sampling frequency since it correlates with the degree of violation of the sampling theorem.

The reconstruction has to be performed on each original pixel but can be parallelized on the GPU and is of $\mathcal{O}(n)$ complexity. This results in a complexity for sampling, contrast compensation and reconstruction of $\mathcal{O}(n/M^2+nk/M^2+n) \approx \mathcal{O}(n)$ and thus, linear complexity.

4.3.3 Automatic Parameterization

The result of a surrogate model can be evaluated after reconstruction by using the convergence threshold as quality threshold (see Section 4.3.1 (p. 149)). Thus, our algorithm computes the perception model on the final result (compensated image) and estimates the quality of the perceived result with the cost functions of the optimization algorithm (see Eq. 3.8 p. 117). If the quality is higher than the quality threshold, we would not perceive any difference between the optimal and the current solution (see Section 4.3.1 (p. 149)). In cases with a quality below the threshold, the sampling interval is decreased and the whole algorithm is restarted to find a better surrogate model. This process of parameter estimation is an optimization process, which can be solved, for instance,

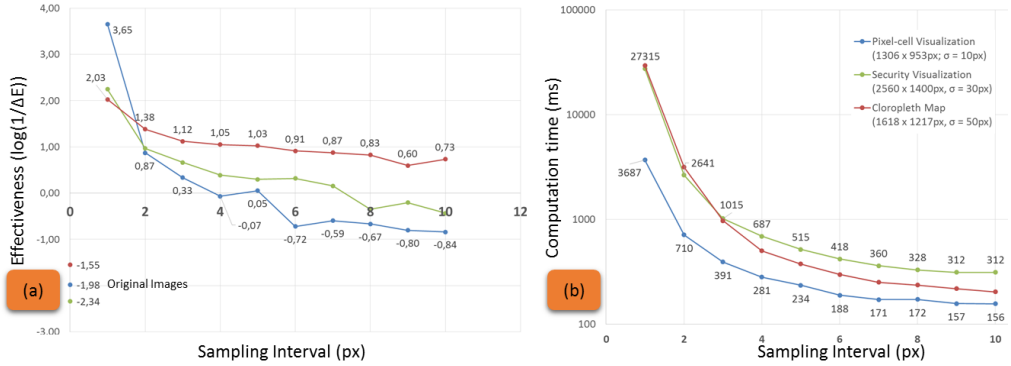


Figure 4.8: Image taken from [11]. *(a) The effectiveness of the algorithm with different sampling intervals and $t = 1_{\Delta E}$ being the quality threshold (zero line due to logarithmic scaling; above zero line, no perceivable difference to the optimal solution). At a sampling interval of 7px, the algorithm is still able to find a "good" solution for the security visualization (green). (b) The computation time decreases significantly with increasing sampling interval. With 7px sampling interval the security visualization can be rendered in 360ms.*

with greedy algorithms or by simply decrementing the sampling interval. Section 4.6 (p. 174) discusses application depended parameterization and provides methods to support the algorithm with references to ease the search for a valid sampling interval.

4.3.4 Computational Evaluation

To exemplify the performance of the algorithm, we applied different sampling intervals on different types of visualizations (Figure 4.9): a security visualization [16], a choropleth map (Vanderbei, 2012), and a pixel-based visualization [7]. All three had different resolutions and required different kernel sizes for the perception model (Fig. 4.8). We performed the computational evaluation on a standard laptop (Intel Core i7-4600U; Onboard Intel HD Graphics 4400).

As described in Section 4.3.3, the effectiveness of the method can be estimated by computing the color differences between perceived result and the original image. The efficiency is measured in milliseconds of computation time. Figure 4.8 reveals that the original images of all applications are suffering from

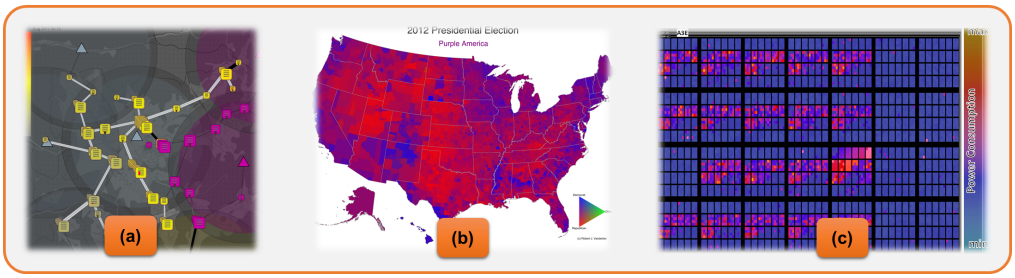


Figure 4.9: *Applications used in the computational evaluation contrast effect compensation with surrogate models. (a) Security visualization taken from [16]. (b) Choropleth map taken from Vanderbei (2012). (c) Pixel-based visualization taken from [7].*

the influence of contrast effects. The method without sampling (equivalent to the method in Chapter 3 but GPU accelerated) is most effective. However, our new method still provides good solutions over the quality threshold, with much higher efficiency. Note, that the security visualization may be sampled with 7px, thereby, reducing the computation time from 27 seconds to 360ms. Since the quality is over the quality threshold, users cannot perceive any difference between this solution and the optimal solution (see Section 4.3.1 (p. 149)).

4.4 Methods for Personalizing Contrast Effect Compensation

Color appearance (including perception of hue, saturation, and intensity) is based on processing red, green, and blue wavelengths of light with the corresponding cone types in the human eye. The original method (Section 3.3.2 (p. 118)) converts the input image into the three LMS channels (for the three cone types). It predicts contrast effects by calculating Eq. 4.3 on each channel, which models achromatic (intensity) and chromatic (saturation, hue) contrast effects. The exponents ϵ_0 (bright center, dark surround) and ϵ_1 (vice versa) control the degree

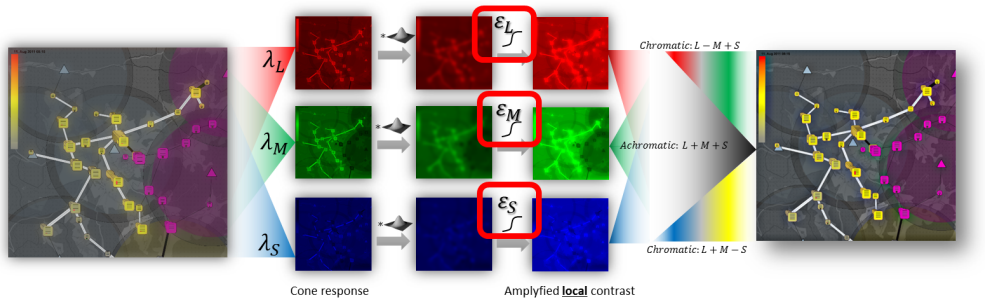


Figure 4.10: *Personalized perception model for predicting contrast effects. The original image is converted from RGB to CAT02 estimating the activation of the cones in the human eye for red, green, and blue light (LMS channels). The perception of the surround is modeled by Gaussian convolution. The eye naturally amplifies contrasts to boosts structural information to detect edges and natural shapes. This is done by amplifying local contrast by the cones that compare center and surround information. Thereby, the perception model predicts contrast effects on the lowest levels of color perception (LMS channels). In the perception model applied in the original method, the same parameter ϵ of local cone contrast is used for all channels. But the channels are different, e.g., we perceive much more structural information in the green channel than in the blue channel because the human eye has much more green cones than blue cones. Further, there are individual differences in spectral sensitivity (Webster et al., 2010) and cone distribution (Curcio et al., 1987). Therefore, the aim of personalization is to find and personalize the parameter ϵ for each of the channels.*

of perceived local cone contrast. The method applies the same exponents for all different types of cones. However, we assume that this is not sufficient and our goal is to determine six exponents ϵ_{X0} and ϵ_{X1}, two for each cone type (e.g., ϵ_{S0} and ϵ_{S1} for blue), to accurately model individual perception of (a)chromatic contrast effects (Figure 4.10). The exponents ϵ_0 and ϵ_1 can then be replaced by the personalized exponents for each corresponding LMS channel. In the following section we present methods of personalization and a user study to validate our hypothesis.

$$PM(I, S, \epsilon) = \left(c_1 \cdot \frac{D65}{c_3 \cdot (S_p / I_p)^{\epsilon}} + c_2 \right) \cdot I_p \tag{4.2}$$

$$PM'(I, S) = PM(I, S, \epsilon), \text{ with } \begin{cases} \epsilon = \epsilon_0, \text{ if } S_p > I_p \\ \epsilon = \epsilon_1, \text{ else.} \end{cases} \tag{4.3}$$

4.4.1 Methods of Interactive Personalization

Method of Adjustment.

We show the user different samples of contrast effects (Fig. 4.11). The samples consist of red, green, blue and gray patches on different backgrounds in order to produce extreme contrast effects in each type of cone and all types combined (gray sample). The left and right patches in each sample as well as the center share the same color (50% intensity in the according channel in the CAT02 color space). The backgrounds are set to zero and full intensity (e.g., black and white). The compensation algorithm is applied on the image and the result is visualized to the user. The slider under each patch controls the exponents ϵ_{X0} and ϵ_{X1} and can be modified by the user. For instance, the left gray patch on black background is perceived brighter (Fig. 4.11(b)). If the user moves the left slider (ϵ_{C0}) to the right, the compensation algorithm will (more and more) darken the gray patch (Fig. 4.11(c)). For personalization, the user tries to match the left gray patch with the gray center and then tries to match the right patch with the center until the patches and the center appear similar. The six personalized exponents are found, if the patches of each sample appear similar.

Staircase Method.

An alternative method to personalize the exponents is the staircase method. For each exponent, we show the user an image with patches (for red, green, blue, and gray, see Fig. 4.12) compensated with different parameters, e.g., for bright centers and green cones: ϵ_{M0}: $\{0.4, 1.0, 1.6\}$ (difference: $\Delta\epsilon_{M0}$: 0.6). The user selects the compensated patch, which appears most similar to the reference, e.g., ϵ_{M0}: 0.4. Then, the parameter is iteratively refined with the halved difference in each iteration until the user does not perceive any difference between the patches, e.g., ϵ_{M0}: $\{0.1, 0.4, 0.7\}$ with ($\Delta\epsilon_{M0}$: 0.3) in the second iteration.

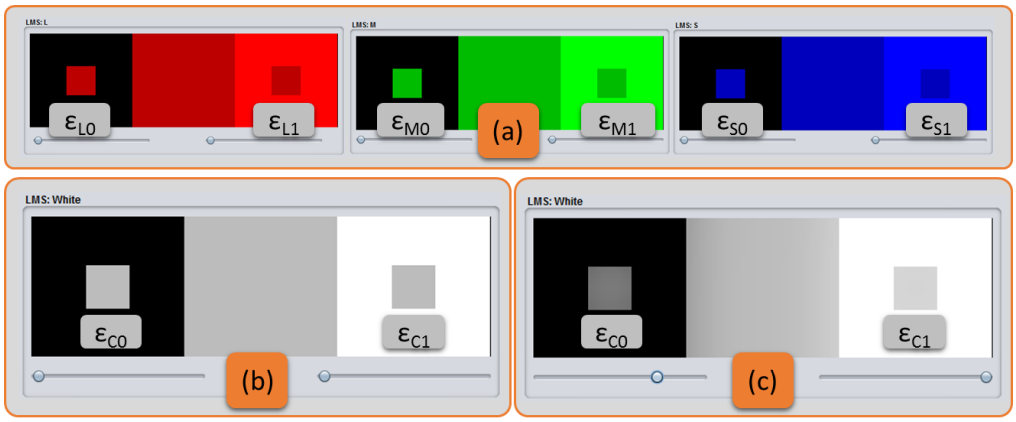


Figure 4.11: Image taken from [11]. *(a) The user is shown samples of contrast effect for each type of cones in the human eye. The colors of the left, right, and middle patch in each sample are equal. The sliders parametrize the perception model. The user is asked to match the colors of the patches with the center and thus, personalizes the method. (b) No compensation with* $\epsilon_{C0} = 0$ *and* $\epsilon_{C1} = 0$*. (c) Compensation with* $\epsilon_{C0} = 0.8$ *and* $\epsilon_{C1} = 1.0$*.*

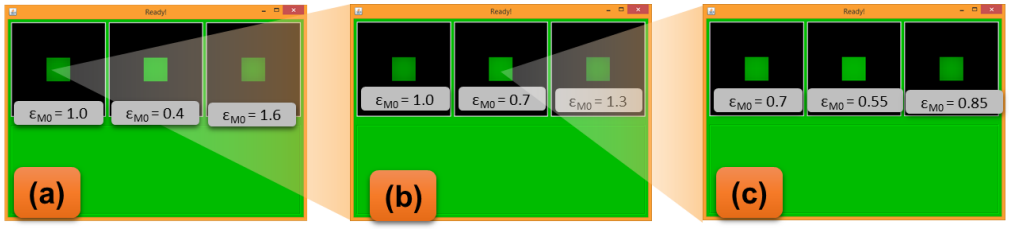


Figure 4.12: *Staircase method for personalization: (a) The three patches should share the same color as the reference color below the patches. The patches are compensated with different values for* ϵ *with* $\Delta\epsilon = 0.6$*. The user selects the patch that appears most similar to the reference. (b) and (c) The parameter is adapted and iteratively refined with decreasing* $\Delta\epsilon$ *until the user does not perceive any difference anymore.*

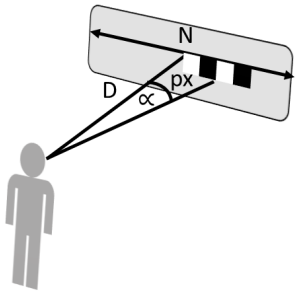


Figure 4.13: *Parameters for adapting view distance and contrast sensitivity functions, with view distance D, display size N, contrast pattern size px, and visual angle α. If the contrast pattern size is too low for our eye to detect (high spatial frequency in degrees α per visual angle), we perceive the average color of the pattern (averaged color at a lower spatial frequency).*

Imbalanced Cone Exponents

There is one issue: imbalanced exponents lead to chromatic shifts. For example, if the exponent ϵ_{S0} for blue cones is significantly higher than the exponent ϵ_{L0} for red and ϵ_{M0} for green, a gray patch will appear yellow on a black background after compensation. Therefore, we balance the exponents for each cone type by pooling the measurements of the colored samples with the measurement of the gray sample (e.g., $\epsilon'_{X0} = 0.5(\epsilon_{C0} + \epsilon_{X0})$). The pooling is repeated until no chromatic shift is perceived by the user.

4.4.2 Contrast Sensitivity and View Distance

The prediction of contrast effects is based on the comparison of center and surround in LMS channels (red, green, and blue light received by the different cone types). As described in Section 3.3.2 (p. 118), the surround is modeled by convolving each LMS channel with a Gaussian kernel. The size of the kernel is adapted to the spatial frequency of the image by measuring the frequencies within the image by difference-of-Gaussian filters of varying sizes. The method, thereby, assumes that each pixel can be visually detected by the observer.

However, the human eye has limits to detect contrast spatially (Schade, 1956). If the frequency of a pattern is too high, we tend to perceive the average color but cannot detect the frequency of the pattern (Xiao et al., 2011). Contrast sensitivity depends, furthermore, on the view distance (Isenberg et al., 2013), age (Crassini et al., 1988), and other degenerative effects (Mei and Leat, 2007). Several models for contrast sensitivity exist surveyed by Watsom and Ahumada Jr (2005). Further, Pelli and Bex (2013) provide guidelines to measure contrast sensitivity. Figure 4.14 shows the CSF of Mannos and Sakrison (1974) as an example.

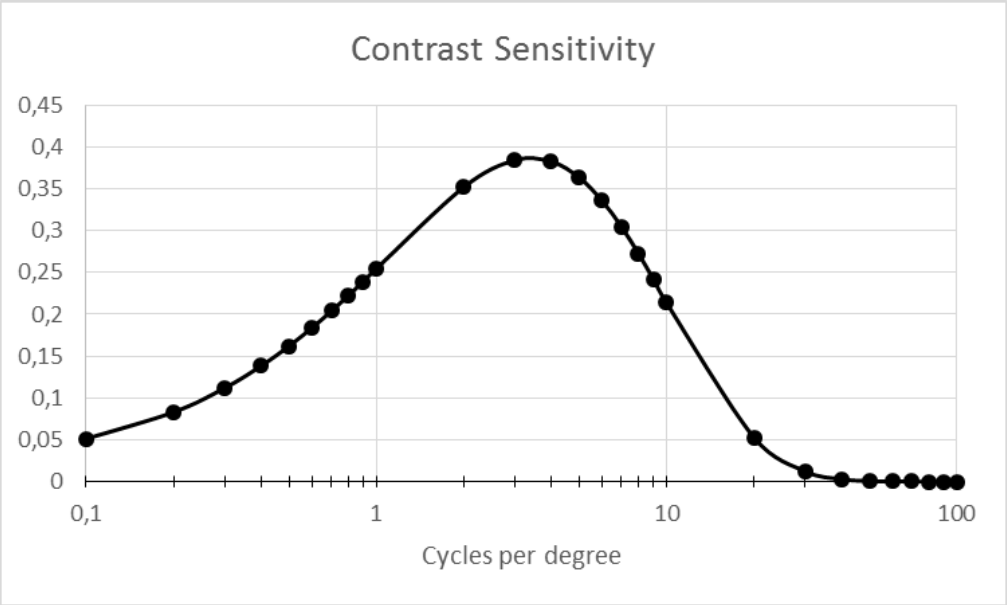


Figure 4.14: *Contrast-sensitivity-functions of Mannos and Sakrison (1974) in cycles per degree of the visual angle.*

Isenberg et al. (2013) used the CSF of Barten (1999) to model contrast sensitivity for different view distances. Following this approach (to adapt display algorithms to CSF), the size of pixels can be converted to degrees of visual angles. For example, on a display with 31cm width N, 1920 pixels in the horizontal n, and 60 cm viewing distance D, a pixel has the size of $0.015°$ of the visual angle (Eq. 4.6). If, for instance, black and white pixels would alternate pixel-wise over the display (pattern size $1 + 1$ pixels), this would end up in a frequency

of $\frac{1}{2 \cdot 0.015} \approx 33$ cycles per degree (Eq. 4.5). According to the CSF (Eq. 4.4, with $a = 0.9846$, $f_0 = 2.4887$, $p = 7748$ (Watson and Ahumada Jr, 2005)), our sensitivity rapidly decreases over 10 cycles per degree ($3 + 3$ pixels) and converges to zero for any frequency above 30 cycles per degree. We, therefore, would not perceive this alternating pattern but just the average (gray) color. The average human eye has a peak-sensitivity for contrasts at 3-4 cycles per degree of visual angle (Isenberg et al., 2013). Patterns between 1.5 and 7 cycles per degree ($22 + 22$ pixels and $4 + 4$ pixels in the example) are still effectively detected with over 75% of the peak sensitivity. Accordingly, another pattern with 10 pixels white and 10 pixels black ($10 + 10$ pixels), which has a frequency of $\frac{1}{2 \cdot 0.15} \approx 3$ cycles per degree, is effectively detected with the peak sensitivity of our eye (Figure 4.14). However, if the observer moves 2m away from the display this results in around 11 cycles per degree and our contrast sensitivity is halved (Figure 4.14).

$$CSF(f) = (1 - a + \frac{f}{f_0}) \cdot exp(-(\frac{f}{f_0})^p) \quad \text{by (Mannos and Sakrison, 1974)} \tag{4.4}$$

$$f(px) = \frac{1}{\alpha(px)} \tag{4.5}$$

$$\alpha(px) = arctan(\frac{N/n}{D}) \cdot px \tag{4.6}$$

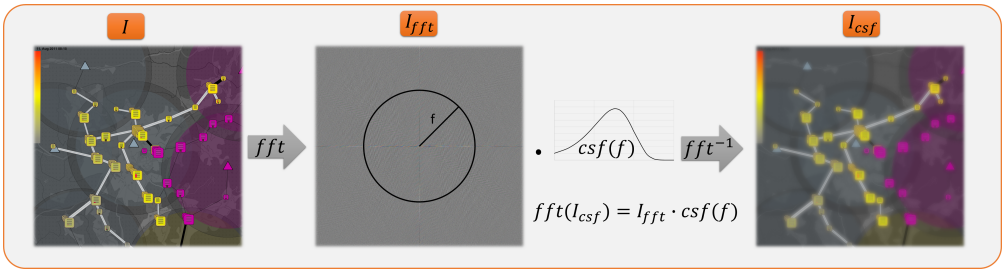


Figure 4.15: *Schematic approach to predict the perceived image according to the view distance. First, the image I is transformed to the frequency domain I_{fft} by Fourier-transformation (in each LMS channel of the CAT02 color space, see Section4.4 (p. 158)). Then, the contrast sensitivity function $csf(f)$ is applied on I_{fft} and transformed to the image domain I_{csf}.*

Using this approach, our method of contrast effect compensation can be made adaptive to the CSF of a user and can integrate the viewing distance (we omit age and other degenerative issues since there is yet no model available). If the viewing distance D of a user is known, the size of a pixel in visual angles can be computed with Eq. 4.6. The input image I can be transfered to the frequency domain by Fourier transformation (see I_{fft} in Figure 4.15). The frequency image I_{fft} can be multiplied with the CSF after the frequency base is converted from cycles per pixel in cycles per degree by Eq. 4.6, which adapts the pixel size to the viewing distance. This removes high frequencies according to the CSF, which are not visible to the human observer. When the image is converted back from the frequency domain, high frequency contrasts are removed and the image I_{csf} does only contain contrasts that are perceived by the human observer (see I_{csf} Figure 4.15). On this result image I_{csf}, we estimate the size of the kernel that models the surround in the perception model with the difference-of-Gaussian method in Chapter 3 (p. 120).

The perception model (Eq. 4.7) can be adapted to integrate contrast sensitivity by replacing input image I by I_{csf}. The rest of the method of contrast effect compensation remains the same. Figure 4.16 (c) shows the result of this new method in comparison to the original method (b). In (b), the contrast effects are reduced globally, which reduces also the contrast of arrows within the (power) lines and the red bar in the bar chart. In (c), the method adapts to the contrast sensitivity and even introduces contrasts to boost the visibility of high-frequency patterns (arrows and red bar). Thus, the high-frequency patterns become more crisp and readable.

$$PM_{csf}(I, S, \epsilon) = \left(c_1 \cdot \frac{D65}{c_3 \cdot (S_{csf,p}/I_{csf,p})^{\epsilon}} + c_2 \right) \cdot I_{csf,p} \tag{4.7}$$

Note, that this method is provided as proof-of-concept for compensation of physiological bias depending on the view distance. Section 4.7 (p. 176) discusses further future work items and the evaluation of the method.

In order to use this approach in a dynamic environment, in which we assume

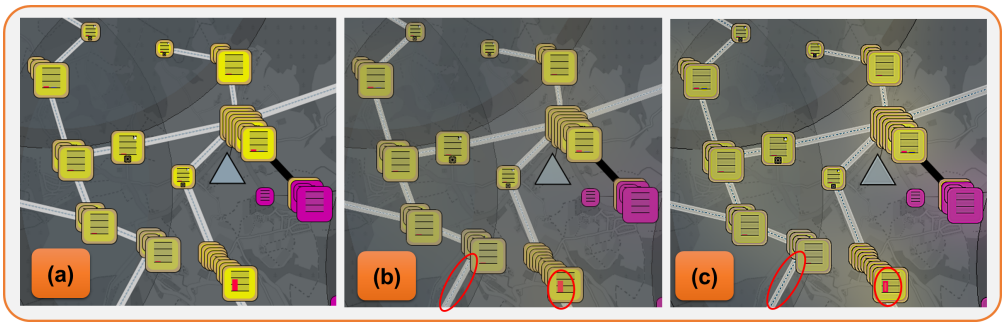


Figure 4.16: *Result of view distance dependent compensation of contrast effects. (a) Original image. (b) Result of the original method. Patterns such as the arrows in the power lines and the red bar become invisible to the analyst due to the view distance and contrast sensitivity. (c) Contrast effects are compensated according to the view distance (4m). While the (big) transformer stations are compensated similar to (b), the small (high frequency) patterns such as the arrows in the power lines and the red bar are boosted to enhance the visibility of these patterns according to the view distance by introducing contrasts.*

that the viewing distance is known but dynamic (e.g., the user is allowed to move), heuristics become obligatory since Fourier-transformation and the compensation algorithm are complex. Therefore, we propose to precomputed the kernel sizes on the application images with several viewing distances and to model these kernel sizes with a regression model (similar to zooming; see Section 4.6.1 (p. 174)). If the user moves, the compensation algorithm adapts the kernel of the perception model according to the regression model. Thus, the algorithm becomes efficiently adaptive to the users distance to the display.

4.4.3 Hardware Dependency and Perceptual Environment Issues

Color appearance is relative and does depend on the observer, but also on the display device and the environment conditions (e.g., ambient light). The iCAM perception model makes the assumption that the display is configured to sRGB (Anderson et al., 1996) and that the ambient light corresponds to the

standardized $D65$ light. Therefore, color calibration plays an important role for the method of compensating contrast effects.

Ideally, the output device is color calibrated and the ICC profile of an output device is available, which enables the transfer of images between two output devices (e.g., LCD display and printer). However, typically images on two devices are not equally perceived since the profile transfer functions or look-up tables are not precise. Further, ICC profiles cannot capture environment issues such as ambient light, aging effects of devices, or malformed configuration.

To address these issues, available hard- and software solutions can be applied, which calibrate display devices and ambient light can be tuned to match the optimal conditions. If the ambient light cannot be adjusted, the perception model can be adapted by replacing the reference white parameter $D65$ to the ambient light (Eq. 4.3). However, this requires determining the ambient light.

For this discussion, we assume that calibration hardware is not available. Todays operator systems and color management drivers for graphic cards and displays offer standard approaches to calibrate the display devices. One method is *gamma correction* that ensures perceptually linear increasing luminance of the display pixels (Brainard et al., 2002). Most of these methods were developed for CRT displays and not for todays LCD displays, for which gamut calibration is more complex (Xiao et al., 2011), however, they are sufficient enough for most applications. There are methods for visual gamma correction with psychophysical experiments, e.g., by To et al. (2013). With both methods it is possible to interactively estimate the limits of the display device (based on their primaries) but also to capture issue with the environment (e.g., ambient light) since they directly measure the perception of the user in his/her environment.

Ambient light and display primaries have strong influence on the appearance of "white" on a display (the reference white). For example, under unusual green ambient light, white appears greenish on the display. To account for influences of ambient light and uncalibrated displays, we follow the approach of To et al. (2013) and measure interactively the reference white in the current environment with a tool illustrated in Figure 4.17. The tool shows patches with increasing intensity for each RGB channel on backgrounds with maximized in-

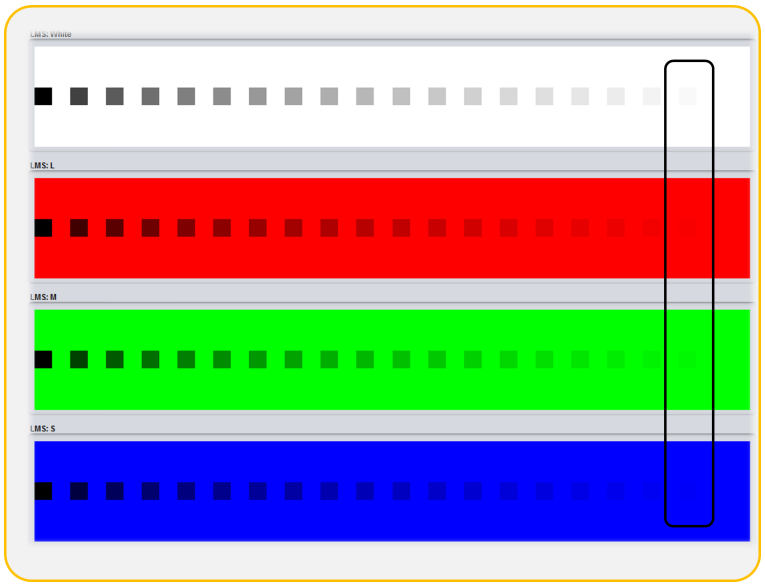


Figure 4.17: *The backgrounds share maximum intensity for each RGB channel. The patches on the backgrounds increase linearly in intensity. The user is asked to determine the patch that is just visible from the background, which determines the maximum intensity for each RGB channel that corresponds to the reference white R of the current environment.*

tensity in the according channel. The user must select the patch that is "just visible" from the background in each RGB channel. Thereby, the user determines the maximum intensity of each channel, which determines the reference white $R = \{R_{max}, G_{max}, B_{max}\}$.

According to To et al. (2013), the relationship of RGB value and output luminance of the display can be modeled with a power function (Eq. 4.8), with I_x being the intensity and R_x being the maximum intensity of the RGB channel x. In the ideal case, the function is linear with $\gamma_x = 1$ that would enable faithful color encodings. However, in practice this case is rare. Therefore, the goal is to determine γ_x for each RGB channel to perform *gamma correction*. Following the approach of To et al. (2013), we determine the mid-point of the power function by the tool illustrated in Figure 4.18. The background pattern is alternating pixel-wise the maximum intensity of each RGB channel R_x and zero. Due to contrast sensitivity, the human eye does perceive the average luminance

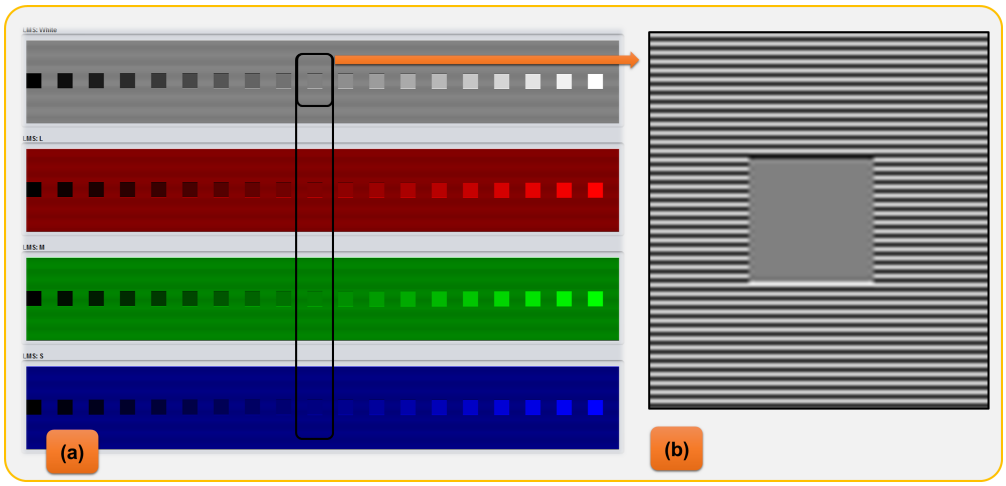


Figure 4.18: *(a) The background for each RGB channel is alternating pixel-wise maximum and zero intensity (b). Due to contrast sensitivity we perceive the average color, e.g. gray in (a). The patches linearly increase in intensity. The user is asked to select the patch that blends best with the background for each channel. This determines the mid-point of the function to perform gamma correction.*

of the pattern, which should be equal to $0.5 \cdot R_x$. The color of patches in the foreground increase from zero to R_x. The user is asked to select the patch that blends best with the background, which determines the midpoint of the power function $t(0.5_x)$. Each γ_x can be determined by Eq. 4.9, which determines the power function of *gamma correction* (transfer function).

Such methods can then be transfered to the perception model for contrast effect compensation by integrating the transfer function t in Eq. 4.10. If methods do not capture the ambient light, the reference white should be estimated with, e.g., the method of To et al. (2013) (described above). The reference white R can then replace the parameter $D65$ in Eq. 4.10.

$$t(I_x) = \left(\frac{I_x}{R_x}\right)^{\gamma_x} \cdot R_x \tag{4.8}$$

$$\gamma_x = \frac{log(t(0.5_x)/R_x)}{log(0.5)} \tag{4.9}$$

$$PM(I, S, \epsilon) = \left(c_1 \cdot \frac{D65}{c_3 \cdot (t(S_p)/t(I_p))^{\epsilon}} + c_2\right) \cdot t(I_p) \tag{4.10}$$

Figure 4.19 illustrates our approach on a uncalibrated display with a skewed green primary. We determined the reference white and performed gamma correction with the tools described above and integrated the transfer function in the method of contrast effect compensation by applying the perception model of Eq. 4.10. The method aims to compensate for low intensities in the green channel by increasing the green intensity. Note, that this method is provided as proof-of-concept for device dependent compensation of physiological bias. A formal evaluation remains future work, see also Section 4.7 (p. 176).

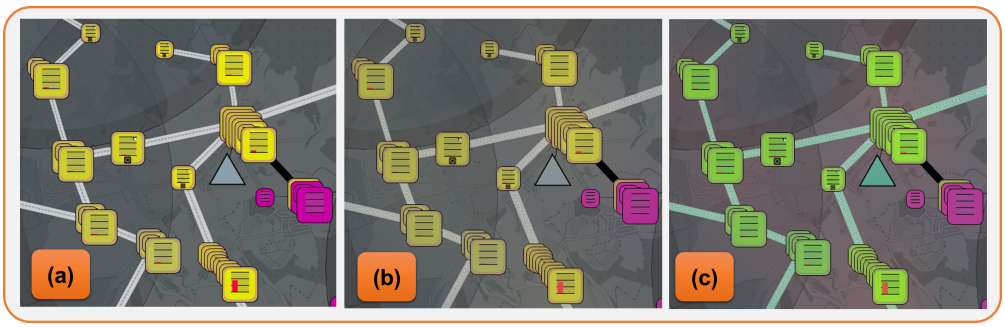


Figure 4.19: *The output display has a skewed green primary (low green intensities). The device dependent method of contrast effect compensation aims to compensate for these issues by increasing green intensities. (a) Original image. (b) Compensated image on calibrated displays. (c) Compensated image for a target (skewed) display.*

4.5 Evaluation of Personalized Perception Models

Chapter 3 showed that compensation of contrast effects significantly improves the accuracy of users reading and comparing color encoded data. The goal of our experiments in this chapter is to evaluate if the perception of contrast

effects differs between subjects, and to evaluate if personalized compensation will outperform our original method. Our hypothesis are the following:

H1 *Subjects find different configurations for personalized contrast effect compensation.*

H2 *Subjects are more accurate in reading and comparing color encoded values with their personalized perception model than with the standard perception model.*

4.5.1 Experiment

The task was to read and compare color encoded data values from displays. The subjects were shown a pixel-cell based visualization with color-encoded time series data and a color legend. The color legend was divided into nine value ranges and the subject was asked to assign two marked data objects to value ranges based on the color legend (see Fig. 4.20). Contrast effect compensation was applied on the visualization with different perception models. The time series was normalized [0,9] and the marked values shared the same data value in the range of [3,7] but had different surrounds. The error in comparison was measured by the difference between the assigned value distance and the real value distance.

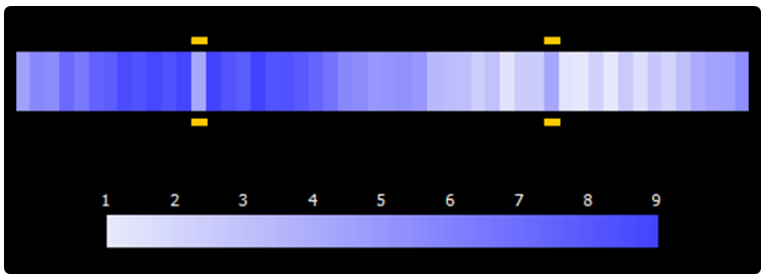


Figure 4.20: Image taken from [11]. *The subjects had to assign the highlighted values to the value range in the color legend (here 4).*

The experiment factors are four different configurations of the perception model applied in the compensation algorithm (personalized model, standard model, the model of another person, and no compensation). Further, three common color legends are selected for each visual channel (see also Figure 3.12 (p. 130)): intensity (grayscale), saturation (blue to white), and hue (blue to red).

Design. The experiment was within-subject designed with 12 subjects (students; 5 females; no visualization experts; normal color vision). Subjects were instructed and performed both personalization methods (adjustment and staircase). The method was selected, which was most satisfying to the subject. Three persons of a pre-study provided their personalized model for the final study. These models were randomly selected as "model of another person". Each subject performed four runs. In each run, the order of (the three) colormaps was randomized and the subjects performed the task with each perception model (four models, order randomized). This results in 12 trials per run and 48 trials per subject. The task was performed under controlled lab conditions with standardized ambient light and a color calibrated monitor (sRGB).

Results. Fig. 4.21 summarizes our results. The Wilcoxon Signed Rank Test was applied for paired significance tests. The median cone contrast exponents for bright centers with dark surrounds ϵ_{X0} was 0.48 (inter-quartile range (iqr): 0.24) as well as for dark centers with bright surrounds ϵ_{X1} (iqr: 0.44). There was a significant ($p < 0.05$) difference in accuracy between personalized perception models (median: 0.96; iqr: 0.35) and the perception models of other persons (median: 1.04; iqr: 0.44). Subjects were also significantly ($p < 0.05$) more accurate with personalized models than with the standard model (median: 1.29; iqr: 0.56). Further, the standard perception model significantly ($p < 0.01$) outperformed no compensation (median: 2.08; iqr: 0.5). As in Chapter 3, there was no difference in accuracy between the colormaps. Five subjects selected the staircase method but there was no difference in accuracy to the subjects that selected sliders.

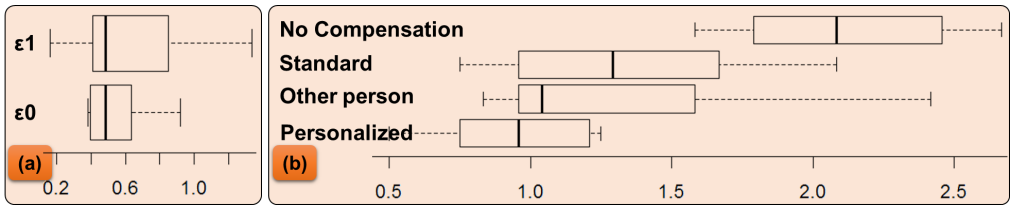


Figure 4.21: Image taken from [11]. *Experiment results. (a) Cone contrast exponents ϵ_0 and ϵ_1 of all subjects. (b) Error of subjects comparing color encoded values with different parameters for the perception model in contrast effect compensation.*

4.5.2 Discussion

The median cone contrast exponents (ϵ_{X0}:0.48; ϵ_{X1}:0.48) of our subjects were close to the standard perception model of Chapter 3 (ϵ_0:0.5; ϵ_1:0.6). However, we can still approve H1. The high inter-quartile ranges reveal that our subjects found different personalized parameters for contrast compensation. Further, there is a significant difference in user accuracy between applying personalized models and the models of other persons in the experiment task. This is not surprising since there is significant individual variability in the distribution of cones (Curcio et al., 1987) and differences in spectral sensitivity (Webster et al., 2010). In our study, contrast effects caused errors in comparing color values of 23%. The error decreased to 14.2% with the standard model and was further significantly reduced to 10% with personalization, which confirms H2. This reveals the need for personalization of contrast effect compensation. However, we found that some subjects had problems with the personalization methods. Five were not able to match the colored patches with sliders. Three had problems to perceive color difference in the staircase method. However, in both cases the subjects had no problems with the other method and there were no accuracy differences between both groups. There exists other methods to capture individual characteristics of color vision (Fairchild, 2013), but it remains unclear, which methods work best for contrast effect compensation.

4.6 Application

In this section, we discuss how our new method for contrast effect compensation can be parameterized and integrated in real systems. We further provide a use case of an existing crisis management system [16].

4.6.1 Application Dependent Parameterization

We provide a configuration tool to personalize the perception model and to parametrize the method for the target application. The parameters for personalization and application dependent parameters (kernel size and sampling interval) are stored in configuration files that can be loaded into the compensation algorithm. One of the application dependent parameters of the perception model is the size K of the Gaussian kernel (Section 4.3.1 (p. 151)), which models the surround perception. This depends on the spatial frequency of the image and on the application. We allow the user to load a reference image of the application and our configuration tool estimates the kernel size by using the approach in Section 3.3.2 (p. 120) that uses Difference-of-Gaussians to determine K. If the application uses interactions, which will significantly change the size of elements, and thereby, the spatial frequency of the image such as zooming, the application should adapt the kernel size. We recommend to load several reference images of different zoom levels. The tool fits the different kernel sizes by determining a regression model, which could be used to adjust the kernel size during runtime if the user zooms. If the application contains different types of visualizations with different spatial frequencies, we recommend to load reference images of all visualizations and to create several configuration files that are used if the analyst navigates from one view to the other.

In addition, our tool can estimate the parameters for surrogate modeling on the reference images. The tool performs compensation with different surrogate

models by increasing the sampling interval and evaluates the performance for each model (see Section 4.3.3 (p. 156)). The tool increases the sampling interval until the surrogate model provides a quality lower than the quality threshold. Then the interval with the lowest computation time but with a sufficient quality is selected. This parameter is stored as reference in the configuration file for the application. This reference may become invalid due to interactions (e.g., zoom and pan). As the compensation algorithm is able to detect if the solutions lacks in quality, parameters can be adjusted on-the-fly (see Section 4.3.3).

4.6.2 Use Case

The monitoring and understanding of the relationship of critical infrastructures and the coordinated management of their failures is one of the biggest challenges in critical infrastructure protection and crisis response. In the application [16], infrastructure elements are represented by symbols, such as rectangles for transformer stations and triangles for mobile stations (see Figure 4.2 (p. 145)). All infrastructures are visualized on a map to reveal their geo-spatial context. Color is used to visualize the status of each infrastructure element, indicating the "request of action". The visual design and especially the coloring is very critical in this domain since this enables decision makers to perceive and understand the situation in its context. Colormaps are, therefore, designed according to the principles and guidelines of Chapter 2 (see also Section 2.3.5 (p. 72)). The solution is a colormap with three color anchors: gray (normal status: no action required), yellow (warning: action required) and red (alarm: immediate action required). The colors between these anchors are interpolated in a perceptual uniform color space varying from gray to yellow over saturation and from yellow to red over color hue with equalized saturation and intensity. This steers the user's attention to critical elements while uncritical elements disappear in the background. Thus, saturation indicates when the operator has to take actions and the color indicates the level of alarm.

A gray scale is used in the background to visualize the geographic informa-

tion and to utilize the effect that uncritical elements blend with the background. Unfortunately, contrast effects boost less saturated colors. This lets analysts perceive elements “more” yellow and more critical than they actually are (see Fig. 4.2 (p. 145)). Therefore, we integrated the efficient method for compensating contrast effects into this application. We personalized and parametrized the method by using our configuration tool (Section 4.6.1 (p. 174)). The application provides overviews for different critical infrastructures such as power grids and street networks. Thus, we had to determine for each view a different configuration file. Further, zooming and panning are the most prominent interactions in this application. Therefore, we determined configuration files for different zoom levels and instantiated a regression model for the parameter K (kernel size) that depends on the spatial frequency of the image.

In our sample application, contrast effects shifted the colors on average by five JNDs (just noticeable differences; see Fig. 4.8 (p. 157): $2^{2.34}JND$). Since the colormap is perceptual linear and provides 40 JNDs, this results in an approximate bias of 12.5% in reading the status of elements on average, which is not acceptable in security applications. The bias is decreased with compensation to only 2% ($2^{-0.32}JND$) on average. The computation time for contrast compensation is decreased from four minutes to 360ms, whereby the surrogate model uses 175ms and the reconstruction 185ms, which is below the latency limit of 500ms for interactions (Liu and Heer, 2014). However, hundreds of milliseconds are still noticeable in seamless transitions such as panning, zooming, or view transition. Therefore, we disable contrast compensation during transitions and perform the method once the image stabilizes.

4.7 Discussion & Future Work

In this chapter, we present an efficient algorithm to compensate for contrast effects and provide a computational evaluation of the effectiveness in comparison

to our original method of Chapter 3. The performance gain makes compensation of contrast effects applicable for interactive visualizations and experiments. We, further, introduce methods to personalize contrast compensation and show in a user study that this significantly improves the accuracy (by 29% in comparison to our original method) of users reading and comparing color encoded data. Although these advances are significant, we see further research challenges and future work to be addressed in this research area.

Interaction & Temporal Context

The runtime of the original method for contrast effect compensation is reduced significantly and we argue that contrast compensation is now possible in interactive applications. However, hundreds of milliseconds are still a burden for dynamic visualizations, animations or seamless transitions such as panning, zooming, or view transition. And further, for complex visualizations in which the rendering and interactions are already computationally expensive. By disabling contrast effect compensation during seamless interactions, we provide an accurate visualization once the image stabilizes. However, we discovered that subjects experience the changes when the compensated image replaces the original image after the interaction is completed. Furthermore, perception of luminance (Eagleman et al., 2004) and contrast has a temporal context (Belmore and Shevell, 2011). It is not clear yet how these effects bias the analyst in the analysis process. Therefore, it remains an open question how to effectively apply contrast compensation during interactions or animations.

Hardware and View Dependent Compensation

In Section 4.4.2 (p. 162) and Section 4.4.3 (p. 166), we introduce methods to compensate for physiological biases based on display primaries, ambient light, and viewing distance. For the proof-of-concept, we provide examples from a security application, which runs in a control room with special hardware for collaborative analysis. In this collaborative application, users are changing their position and their view distance. We performed an informal user study with three participants reading the visualizations with and without the proposed

methods of hardware and view dependent compensation. The results show an increasing trend in accuracy. However, beside this promising proof-of-concept, a formal user study for the evaluation of the proposed methods remains future work.

Further, the presented methods have their limits, for example, the viewing angle (Parraga et al., 2014). The methods are also based on the primaries of the display and cannot capture malformed gamuts of displays. A high potential solution to this problem is presented for a different domain: The work of Flatla and Gutwin (2011) proposes situation-specific models of color differentiation. They are able to measure the effects of color vision deficiency on perceptual uniform color spaces by psychophysical experiments with the target user. Further, their methods can adapt visualizations and displays for the target color blind user. We see high potential in their methods to capture malformed gamuts of display devices and ambient light by interacting with the user resulting in accurate color transfer functions that could be used in the perception model.

Personalization & Perception Model

With our configuration tool it is possible to interactively determine all parameters for the algorithm and to personalize the perception model on the lowest levels of color vision (for each cone type). While the mechanisms of the lowest levels of color vision are well researched and characterized (Fairchild and Johnson, 2004), the mechanistic description of higher levels is still controversially discussed (Stockman, 2009). Some studies conclude that research fails to provide evidence for higher order mechanisms in color vision at all (Eskew, 2009). This indicates that personalization on the lowest levels might be sufficient to capture individual's characteristics of (a)chromatic contrast perception (Ekroll and Faul, 2009). Our experiment shows that this approach significantly improves the accuracy of users reading and comparing color encoded data. However, recently published results present evidence for higher order mechanisms (Hansen and Gegenfurtner, 2013). We see high potential in these results for our method, but it is not clear yet how to transfer these results into a computational model for image color perception. Another interesting hypothe-

sis of Granzier et al. (2012) is that individual differences of chromatic induction result from individual eye movement. While they found no evidence for settings in which the user is allowed to freely look at the display, they found a significant individual difference under forced viewing conditions. This indicates that for some visual analysis task (e.g., focusing on certain elements on the screen for monitoring) individual differences are more critical than in others (e.g., freely exploring the whole data display). It would be interesting to compensate for such effects depending on the analysis task as well, when models for this effect become available.

Collaborative Environments & Augmented Reality

We see another challenge for collaborative environments since perception of contrast effects differs significantly between individuals. The straight-forward solution is to apply the standard perception model. However, our study showed that the error of users reading color encoded data can be reduced by one third in comparison to the standard perception model. Therefore, enabling personalized visualizations in collaborative analysis is a desired but challenging research goal since our experiments showed that the perception model of one person can bias the readings of another person. Practical solutions can create several view spaces such that each analyst has an own personalized view space. Views to communicate findings are then compensated with the standard model. Recently, the research on head mounted device and augmented reality presented interesting and probably ground breaking research results. For example, Hwang and Peli (2014) provide methods for edge enhancement for head mounted devices. Similar, Itoh and Klinker (2015) and Giusti et al. (2011) propose methods for defocus correction by head mounted displays or microscopes. Avery et al. (2009) improve spatial perception for augmented reality x-ray vision. These advances show that head mounted devices and augmented reality can be used to improve the perception of visual data. We see great potential and interesting research challenges in these technologies for personalized visual analysis.

5

Exploiting Contrast Effects for Visual Boosting

Contents

This chapter is based on the following publications and parts of the sections also appeared in the following publications:

[15] Sebastian Mittelstädt, Andreas Stoffel, Tobias Schreck, and Daniel A. Keim. Analysis of Local Data Patterns by Local Adaptive Color Mapping. Presented at the IEEE Conference on Visualization (poster paper), 2014.

[7] Halldór Janetzko, Florian Stoffel, Sebastian Mittelstädt, and Daniel A. Keim. *Anomaly Detection for Visual Analytics of Power Consumption Data.* Computer & Graphics, 38:27–37, 2014.

[18] Lin Shao, Sebastian Mittelstädt, Ran Goldblatt, Itzhak Omer, Peter Bak, and Tobias Schreck. StreetExplorer: Search-based exploration of urban street networks. Submitted to the International Conference on Information Visualization Theory and Applications, 2016.

For the division of responsibilities and work, as well as a statement of contributions in these publications, please refer to Section 1.5 (p. 18).
The following contributions go beyond the published work:

1. Problem definition of *local adaptive color mapping*.
2. An unsupervised *local edge preserving color mapping* algorithm that locally adapts the data values in order to enhance the visibility of low-contrast structures. Thereby, the algorithm preserves, however, the continuous trends and natural forms within the data, without relying on prior knowledge (unsupervised algorithm).
3. A computational evaluation of the *local adaptive color mapping* algorithms in comparison to the state-of-the-art.
4. Use cases for *local adaptive color mapping*.

Chapter Abtract & Structure

The preceding chapters discussed methods for *effective* color encoding and how to compensate for contrast effects in order to increase the accuracy of users reading and comparing data values. In this chapter, however, we present how contrasts can be used to enrich (dense) data visualizations that use color for encoding metric quantities.

Several global color mapping schemes exist that support certain analysis tasks (Chapter 2). However, static global schemes map data with a small local variation (within a data set of high variation) to small color differences. Often, these color differences are below the noticeable difference threshold of human perception or the display device. As a consequence, valuable information may be lost since data points or structures cannot be adequately perceived and correlations or other patterns of interest may be missed. Existing techniques to avoid this effect either require user interaction or are based on specific assumptions about the data.

This relates to Chapter 4, in which high-frequency information is not visible to the observer due to low contrast sensitivity if, for example, the user moves away from the screen. The method presented in Section 4.4.2 (p. 162) can overcome this issue since it introduces contrasts in these cases enhancing the visibility of high-frequency information. However, this method does only consider the bias of our perception and not the bias of the mapping scheme. The method of contrast effect compensation is, therefore, not able to enhance the visibility of local data patterns that are lost due to the color mapping, because the global mapping schemes is the base for the quality metrics and the method tries to satisfy the global mapping, although it may mask local data features.

In this chapter, we introduce novel automatic algorithms for *local-adaptive color mapping* that is applicable to dense data and is based on the idea to locally modify the color mapping to enhance the visibility of data structures that are lost due to global mapping schemes and perceptual bias. Thereby, the algorithms are exploiting contrast effects to enrich data visualizations. The algorithms emphasize patterns of interest within locally chosen color-ranges such that (1) the

visibility of local differences, important data points, and visual structures are enhanced, however, (2) the introduced global distortion of the color mapping is kept small. This allows the perception of relevant patterns while approximately maintaining global comparability across the whole data set.

In this chapter, we motivate this controversy and problem in Section 5.1. Related work is discussed in Section 5.2. The problem of supporting synoptic localization, -identification, and visibility of structures is defined in Section 5.3. Further, this section provides novel automatic algorithms for *local-adaptive color mapping* that visually boost local data structures and enhance the visibility of important data points. Use cases and heuristics for *local-adaptive color mapping* are provided in Section 5.4. Section 5.5 concludes the chapter with discussion and future work.

5.1 Motivation

The pixel-based visualization paradigm is well-known for its effectiveness in visualizing large volumes of data (Keim, 2000). The general idea is to have each pixel encode one data value, by appropriately mapping data values to color. Pixel-based visualizations are highly scalable, which exploits modern high resolution displays. Pixel-oriented techniques can be directly applied to map sequential data to pixel rows. They can also be embedded within other visual designs, e.g., forming glyphs or enhancing traditional diagrams, such as bar charts by an additional level of data detail (Keim et al., 2002). Pixel-oriented techniques have been successfully applied in many analysis domains, including document analysis (Keim and Oelke, 2007), time series analysis (Weber et al., 2001) or analysis of multivariate data (Keim et al., 2002).

One reason for the success of pixel-oriented techniques is the fact that color is, after position, among the most effective visual variables to encode information (Treisman, 1985). In pixel-oriented visualization the choice of color map-

ping should ideally allow reading and comparing data values on an absolute scale, as well as allow comparing data values across the global and across the local scale (see Section 2.3.1 (p. 44)). Since there are many ways to map values to color, this is a critical point in the technique. Given we have a perceptually linear color-map, the dynamic range of input data values may render local data variation unnoticeable due to limits of the human visual perception or display device technology. Patterns in data below a threshold, so called *just noticeable differences* (JND), will not be perceivable by an analyst, and potential relevant information may get lost (Figure 5.1).

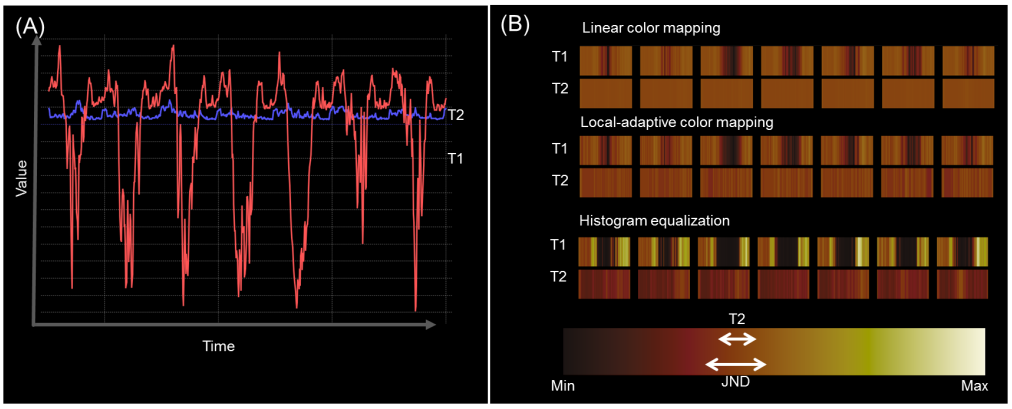


Figure 5.1: *Boosting of peak points in pixel-based time series visualization. (A) shows two time series in a line chart. (B) illustrates the pixel-based visualization of the same data with different color mapping schemes (each rectangle holds 24 hours of data). The trend of T2 is invisible in the linear representation, which is recovered by the other color mappings. However, the non-linear histogram-based algorithm transfers some value ranges that were orange into the yellow and white ranges of the color map, which biases global comparison.*

Assuming we can identify the variations within the data, we may be able to define an appropriate color scale for the particular case, which embeds an adapted *scaling* of data values. For instance, square root normalization spreads data concentrated in lower intervals of the input data domain to a larger region in the color-scale. It, thereby, improves the distinction of data values in this area. Other mapping schemes including logarithmic or histogram-based exist that can emphasize data concentrated in specific intervals. They can support the analysis of local data properties of interest. However, such techniques re-

quire *a priori* understanding of the data and its relevant scales. They are static; thus, they apply one scheme to the whole data set. And they, further, introduce non-linearities that harm the comparison on the global and absolute scale (comparison requires perceptual linearity, see Section 2.3.1 (p. 44)). For example, in Figure 5.1 (A), the periodic patterns of time series T1 and T2 are visible in the line charts. However, the patterns are invisible in the linear pixel-based display (Figure 5.1 (B)). If we apply a histogram-based algorithm, the trend of T2 becomes visible. However, it maps the original orange color tones into yellow and red. If we want to compare these values, one would intuitively say, that T1 is far higher than T2. Indeed, the averages are about the same.

Similar to the problem of contrast effects in Chapter 3 and Chapter 4, color mappings yet do not consider the surround of pixels in visualizations and cannot guarantee the visibility of structures in the final visualization. This is one of the general problems of visualizations, for example, to enhance the visibility of structures such as streets on maps or veins in the human body but also in abstract data visualizations where sets of pixels form important structural information or visual patterns.

5.1.1 Contributions

Given these shortcomings of static color-mapping schemes, we propose two novel algorithms to dynamically assigning color to dense data regions, based on local data properties. These, so called *local-adaptive color mapping* algorithms, retain the visibility of local details without using additional visualization space and aim at keeping the global distortion small. For example, in Figure 5.1 (B), our *local-adaptive color mapping* reveals the variations of T2 that are masked by linear mapping, however, does not distort the global color scheme as the histogram-based approach. The technical achievements are as follows:

1. An unsupervised *local edge preserving color mapping* algorithm that locally **adapts the data values** in order to enhance the visibility of low-

contrast structures. Thereby, the algorithm preserves, however, the continuous trends and natural forms within the data, without relying on prior knowledge (unsupervised algorithm).

2. A supervised color boosting algorithm that locally **adapts the color mapping** for important data structures and guarantees the visibility of important data points, based on just noticeable differences.

3. Both methods preserve global metric quantities of the data and provide an informative overview without interaction.

4. Heuristics for *local adaptive color mapping* for boosting of important data points and contrast enhancement in high frequency visualizations.

5. A computational evaluation of the *local adaptive color mapping* algorithms in comparison to the state-of-the-art.

5.2 Related Work

Bertini and Santucci (2006a) originally formulated the problem that features and patterns in the data are masked by the visual mapping and the perception of a user. They provided a framework (Bertini and Santucci, 2006a) to measure "what is in the data", "what is perceivable by the user", and how to make data features visible. For scatterplots Bertini and Santucci (2006b) provide visual quality metrics for visual overplotting, provide sampling strategies to overcome this issue, and evaluate their methods with user studies.

Our method can, therefore, be considered as another instantiation of the framework of Bertini and Santucci (2006a) for color mapping in dense data visualizations. Our *color boosting algorithm* measures "what is perceivable by the user" and "what is in the data". If data features are not visible (due to the

color mapping), it uses visual boosting in order to enhance the visibility of these features.

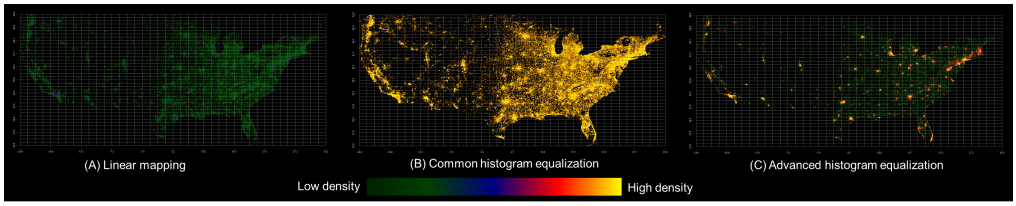


Figure 5.2: *Comparison of histogram-based algorithms.* ***(A)*** *illustrates the rendered image of a density data set, created by linear color mapping.* ***(B)*** *shows the rendering with common histogram equalization.* ***(C)*** *shows the result of the method of Bertini et al. (2007), which we refer to as* histogram-based *algorithm in this chapter.*

Approaches that use histogram equalization optimize the visual distinction of data points by equalizing the distribution of the color scale in the image, e.g., Herman et al. (2000) and Pizer et al. (1987)). An example of advanced histogram equalization of Bertini et al. (2007) is shown in Figure 5.2 (C). This algorithm produces a scheme, which maps the data to perceptual uniform distant color levels (we refer to this technique, if we mention "histogram-based algorithm" in this chapter). The most effective results will be produced, if only few distinct data values are present that are non-uniformly distributed over the value range. The major disadvantage is that the global comparison of values is biased, due to the non-linear mapping, and that it is hard to mentally re-construct the data values as Figure 5.1 illustrates.

In volume rendering, several approaches have been established for enhancing the visibility of isosurfaces within volume data. Local histograms are used in (Lundström et al., 2006) to enhance the visibility of local structures, requiring the user's domain knowledge. Bruckner and Möller (2010) present an automatic method to compute the similarity of isosurfaces. The result is visualized in a two dimensional similarity map where patterns assist the analyst to select a meaningful subset, which is then highlighted in the rendering. Ruiz et al. (2011) present a framework to create transfer functions on the basis of target distributions. This can either been done by providing domain knowledge or by an automatic global distribution, such as in the histogram based

methods discussed above. Wang and Kaufman (2012) present an approach for segmenting volumetric objects by involving an automatic approach for color selection, color attentiveness and color harmony. These approaches are either limited in the number of objects that can be separated (e.g. six in (Wang and Kaufman, 2012)), need interaction and/or have drawbacks that come from globally defined transfer functions (they do not use local structure information).

A more general approach to emphasize local structures would be to add a layer of information on top of the base visualization. Interesting regions can be marked by a layer of emphasis as an entry point for the analyst who may initiate further zooming (Perlin and Fox, 1993). Layering works best if used carefully and is not overdone (Tufte, 1990). If there are too many interesting areas, the marking will lose its effectiveness, since the analyst may get overloaded. Thus, highlighting or layering may not be the best choice to solve our problem of visualizing small invisible local differences. Also, lenses (Elmqvist et al., 2011) could be used as an interactive tool to emphasize local visualization aspects, but here, it depends on the user to actively spot areas of interest in the data.

Oelke et al. (2011) discuss several approaches to enhance the visibility of data points, so called visual boosting. *Boosting by halos and color* visually highlights interesting data points or borders by changing the color of their surrounding pixels, in order to increase the contrast. In detail, *boosting by color* overwrites empty pixels around the interesting data points. The boosting techniques presented by Oelke et al. (2011) are applicable predominantly to sparse data. In this chapter, we consider dense data. Therefore, these approaches are not ideal as we want to preserve the metric quantities of the data and avoid overwriting of quantities. Thus, our algorithms can be interpreted as an additional case for visual boosting for dense data.

5.3 Algorithms for Local Adaptive Color Mapping

In the following, we define the problem of *local adaptive color mapping* and provide two solutions: a supervised *color boosting* algorithm based on just-noticeable-differences and an unsupervised *edge preserving color mapping* algorithm.

5.3.1 Problem Definition

Data Patterns and Visual Structures

In our definition, *data patterns* are application depended patterns in the data that are mapped to visual encodings. They contain valuable information and are of high importance for the analyst. The methods presented in this chapter aim to visually boost *known* data patterns. Data patterns consists of one or multiple *visual structures*, which are application independent. *Visual structures* comprise sets of connected pixels where the spatial configuration (structure) forms interpretable information of the data set. Boosting visual structures will boost the according data pattern as well. Examples are pixels that encode the same time interval of a time series, or a gradient curve in a visualization, rivers on map representations, or a single pixel-cell encoding an important data value.

Visibility of Visual Structures

In Figure 5.3, the curve is only partially visible since on the left side, the color difference between the curve and the background is less than the just-noticeable-difference (JND).

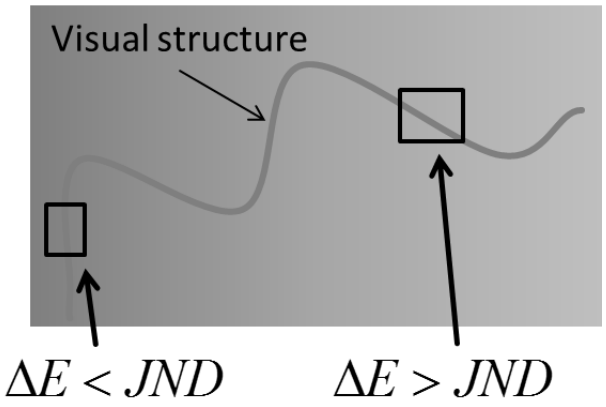


Figure 5.3: *The visibility of a structure depends on its surround. If the color difference ΔE is higher than the just-noticable-difference (JND), the human visual system is able to detect the structure.*

A *data pattern* D is visible, if all visual structures S of D are visible (Eq. 5.1). Accordingly, a structure S is visible, if all its border pixels are perceptually visible (Eq. 5.3). This implies that the color distance of each border pixel p must be greater than the JND to its surrounding pixels n, excluding the neighboring pixels of the same structure. However, the color difference must be smaller than the JND to the surrounding pixels p_s of the same structure (Eq. 5.2) since the analyst should recognize the pixels as a set of connected pixels.

$$visible(D) : \forall S \in D | visible(S) \tag{5.1}$$

$$visible(S) : \forall p \in borderPixels(S), \forall p_s \in S \; | visible(p) \wedge \Delta E(p_s, p) \leq JND \tag{5.2}$$

$$visible(p) : \forall n \in neighbors(p) | \Delta E(n, p) > JND \tag{5.3}$$

However, to be formally correct, the environmental settings and perceptual issues should be considered. Therefore, it is advisable to predict how the observer perceives the image. A personalized perception model PM that considers the output devices, ambient light, viewing distance, and the perceptual issues of contrast effects and contrast sensitivity is presented in Chapter 4. For accurate measures, PM should calculate how the input I and locally adapted image I'

is perceived and then calculate the visibility of structures (and color distortion, see below) in $PM(I')$. However, the human eye tends to amplify contrast for detecting structures in images (Ware, 1988). Therefore, we can assume that if a structure is just noticeable from the background in I', measured with our naive metric, our perception will further amplify these contrasts. Thereby, the structure will be "even more" visible. Therefore, estimating the visibility of structures by Eq. 5.2 is sufficient for most applications and displays, assuming a standard color calibrated display, $D65$ ambient light, and standard viewing distance ($\approx$ 60cm). If these standard conditions are not valid for the target system and observer, the method in Section 4.4 (p. 158) can be applied after the methods in this chapter as post-processing step to personalize the perception model and to compensate for different viewing distances and ambient light. Details on how to combine the algorithms of local adaptive color mapping in this section with the method of contrast effect compensation are provided in Section 5.5 (p. 218).

Optimization Problem of Visual Boosting

In Figure 5.4(a), the color difference to all border pixels is higher than the JND. Thus, this part of the curve is visible. In Figure 5.4(b), the structure visibility is violated since the color differences to the border pixels are very low.

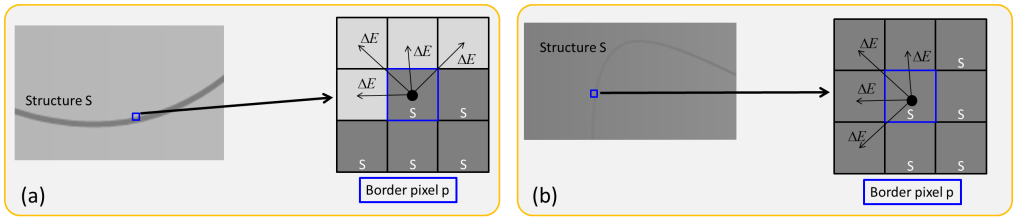


Figure 5.4: *Structure visibility. (a) The color difference between the border pixels of structure S and the background is higher than the JND; thus, the structure is visible. (b) The structure is hardly visible since the color difference between the border pixels and the background is smaller than the JND.*

In order to make the curve globally visible, the color difference can be maximized by assigning it the color black. However, this would distort the original color encoding and bias the user reading and comparing color encoded data val-

ues. Therefore, colors should only be distorted as minimal as possible to allow accurate readings. To measure the amount of color distortion of a *local adaptive color mapping* L_{CM}, the color distortion CD is calculated by the difference to the baseline color mapping B_{CM} in a perceptual uniform color space such as DIN99 (see Section 1.6.1 (p. 24)):

$$f_{CD} = \sum_{i \in \{S\}} CD(i) \tag{5.4}$$

$$CD(S) = \frac{1}{|\{S\}|} \sum_{p \in S} \Delta E(L_{CM}(p), B_{CM}(p)) \tag{5.5}$$

with $|\{S\}|$ being the set of visual structures, p being the pixel of structure S, and ΔE measuring the color difference between the mappings $L_{CM}(p)$ and $B_{CM}(p)$ that assign p a color.

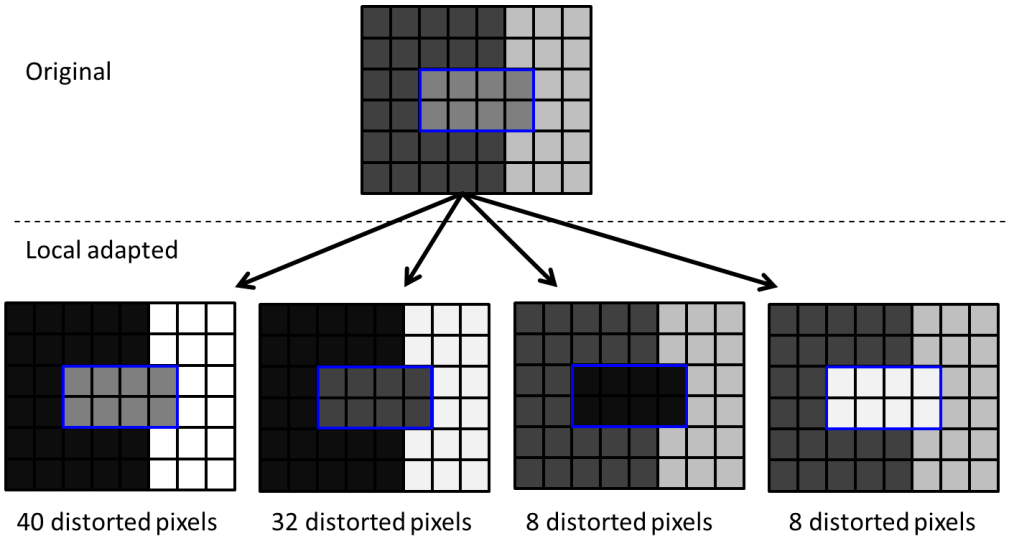


Figure 5.5: *Different valid solutions to make one visual structure visible that is surrounded by two other structures. While solutions (a) and (b) distort the other structures and distort a high number of pixels, (c) and (d) distort only a few pixels. However, it is not clear whether this local optima is the global optima since the structures may be surrounded by other visual structures as well.*

The goal of color boosting is to make as many structures visible as possible. Therefore, the goal can be formulate by the sum of visible structures (Eq. 5.6)

with Φ indicating the importance of a structure for the analysis.

$$f_{VIS} = \sum_{i \in \{S\}} \begin{cases} \Phi, \text{ if } visible(S) \\ 0, \text{ else.} \end{cases} \tag{5.6}$$

However, numerous solutions are possible in order to make a structure visible. For example, Figure 5.5 shows three neighboring structures and four valid solutions to make the structure in the middle visible. Although (c) and (d) distort less pixels than (a) and (b), it is not guaranteed that this local optima is also the global optima since these three structures may also be surrounded by other structures. Further, changing the appearance of one structure changes the perception of other structures, which may, thereby, become invisible.

For s structures and l color levels, the number of possible solutions is l^s. An algorithm must find a solution that maximizes the visibility of visual structures (Eq. 5.6) and minimizes the color distortion (Eq. 5.4) at the same time. Therefore, *local adaptive color mapping* is an optimization problem of two conflicting goals.

5.3.2 Color Boosting based on Just-Noticeable-Differences

This section introduces one solution for the problem of *local adaptive color mapping* by supervised color boosting based on just-noticeable-differences. Our alternative method, an unsupervised edge preserving color mapping algorithm, is introduced in the next section.

We illustrate the basic idea of our algorithm by the example in Figure 5.6. On the left side in Figure 5.6, the structure is not visible because of the low color difference to the background. Therefore, we locally manipulate the color mapping in order to make these details just noticeable. Our color boosting algorithm performs four steps in order to reveal local structures:

1. Structure detection: Local structures of interest are detected by a generic image-based analysis method, or an application dependent detector.

2. Filtering: Noise and irrelevant structures are filtered out.

3. Sorting: The visual structures are sorted according to their *importance* as a heuristic for solving the optimization problem and maximizing Eq. 5.6.

4. Color boosting: The color value of these structures is scaled in both directions of the color map until they become just noticeable from their spatially surrounding area, minimizing Eq. 5.4.

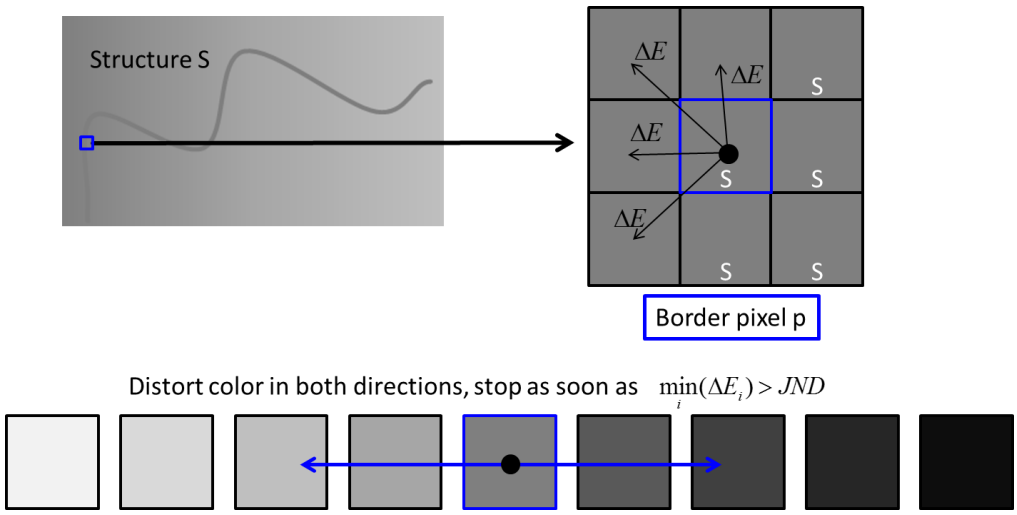


Figure 5.6: *The color boosting algorithm locally adapts the color levels of a structure in both directions of the colormap until the structure becomes visible.*

In the one-dimensional case of Figure 5.1 (p. 184), each structure can have at most two neighboring structures (left and right). However, in other applications each structure can have several neighboring structures. Making all structures perceptual visible has in some cases no optimal solution, since the boosting of one location can decrease the visibility of the adjacent structures. Therefore, we prioritize the structures to be boosted by a measure of their importance to maximize Eq. 5.6. This importance function is application dependent and may be adjusted based on data or image properties.

Two dimensional visualizations, such as maps, scatterplots or any other visualization on a 2D display, contain more complex structures (Figure 5.7). To make these perceivable, the appearance and requirements for structures have to be defined for each application before the actual boosting. A general approach is to group connected pixels that share the same color value and test these groups for structure requirements (e.g., round shape, size, values). However, an unlimited number of requirements are possible. Therefore, the structure detection part of our color boosting algorithm can be replaced by any domain specific method.

Structure detection

As mentioned before, the structure detection is a modular function within the algorithm and can be replaced by any meaningful approach. In this section, we want to exemplary present a variant that is based on edge detection as an application independent approach (application dependent approaches are demonstrated in Section 5.4).

This image based algorithm does not require domain knowledge or assumptions about the data. The input image may be rendered in any pixel-oriented visualization technique with a linear color map that is perceptually uniform and follows the guidelines of Chapter 2. For example, the heat map of Figure 5.1 varies over hues from black over red and yellow to white with increasing intensity, which is done in the DIN 99 color space.

The image-based detector applies the common Sobel operator (Duda et al., 1973) to the linear representation and stores the result E. In order to assign pixel to edges, the algorithm computes the median of all values in E. All pixels with a higher value than the median are tagged as edges (Figure 5.7). We then use these local interest points as seed points for the region growing algorithm (Adams and Bischof, 1994). Each region grows from the seed point, adding pixels from the 8-neighborhood with the same color value, until one structure is segmented. The identified structures are now tested for size, and we omit structures smaller than

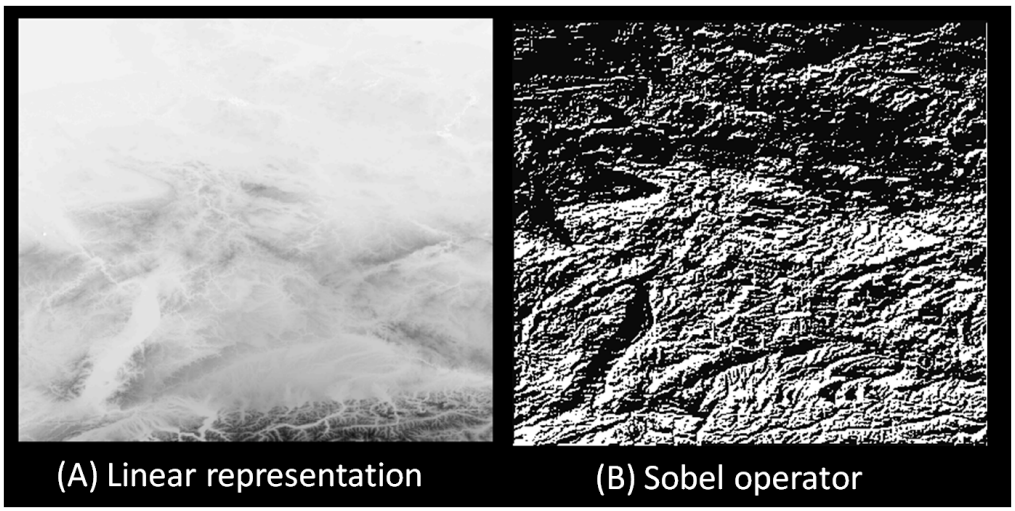


Figure 5.7: *Image-based Structure Detection. (A) Original image. (B) Detected structures with Sobel operator (Duda et al., 1973).*

five device pixels. The latter can also be considered a user- or device-dependent parameter.

There are many more methodologies for detecting seed points and segmenting areas that we could use from image processing (Gonzalez and Woods, 2007). For example, it may be useful to consider also the shape of structures or even better, search for certain shapes or patterns in the visualization, such as parallel stripes in recursive patterns (Keim et al., 1995), which indicate a periodic structure. Note, that this part of our color boosting algorithm is exchangeable and can be replaced by other detectors as needed.

Color boosting

Maximizing the visibility of all structures but minimizing the color distortion is an optimization problem. Our technique solves this optimization problem by sorting the structures in a priority queue according to their importance for the current visualization and then sequentially processes each structure. The list of structures is sorted descending to their size in a priority queue, since

we consider bigger structures to be more (visually) important. This has two advantages: First, important structures are boosted after non-important ones, which ensures to a high probability that the most important structures are visible (maximizing Eq. 5.6). Second, due to the boosting of the smaller structures beforehand, the boosting of bigger structures might be omitted since they might be already visible. This also optimizes our criteria to keep the global distortion at a minimum (minimizing Eq. 5.4).

After the important structures have been identified and sorted according to their importance, the color boosting algorithm remaps the color to preserve their visibility according to the just noticeable difference. The algorithm aims to maximize the number of visible border pixels of a structure, while minimizing the bias of the boosting. This (sub-)optimization problem can easily be tackled by an algorithm, that increases and decreases the values of the structure pixels by Δb and tests in each iteration if the structure has become visible (see Algorithm 1). A meaningful setting for this parameter is $\Delta b = \frac{1.0}{\text{\#colorLevels}}$, with pixel values normalized between 0 and 1 and $\#colorLevels$ being the number of differentiable color levels in the color map. As soon as enough border pixels of a structure are visible (thereby, the whole structure) the algorithm stops and continues with the next structure in the priority queue. As a rule of thumb, we set this to 99% of all border pixels of a structure. According to Eq. 5.3, a border pixel is considered as visible, if the color distance of the pixel to every of its surrounding pixels is above the JND threshold (Algorithm 2). Color distances are estimated in the DIN 99 color space, where $\Delta E = 1$ approximates the JND (see (Mahy et al., 1994) for accurate measurements). Different parameterizations are illustrated in Figure 5.8.

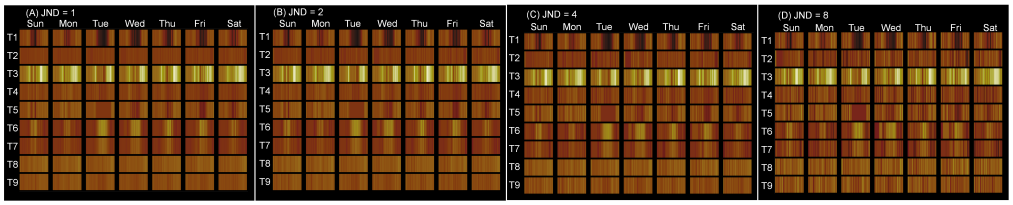


Figure 5.8: *Results of our color boosting algorithm with different parameter settings for the JND.*

This algorithm has a complexity of $O(s * p * w * l)$ where s is the number of structures, p the number of pixels per structure, w is the number of neighbors (standard: 8 neighborhood) and l the number of distinguishable color levels. Since, in the worst case $s * p = n$ (number of pixels) and $w, l << n$, the algorithm has a linear complexity of $O(n)$.

```
Data: Structures sorted according to their importance
Result: Boosted structures
for Structure s in structures do
    high = s.copy();
    low = s.copy();
    while !visible(high) && !visible(low) do
        for i=0; i < high.size; i++ do
            high[i].value += Δb;
            low[i].value -= Δb;
        end
    end
    if visible(high) then
        visibleStructures.add(high);
    else
        visibleStructures.add(low);
    end
end
return visibleStructures;
```

Algorithm 1: Guarantees the visibility of important structures. The variable *high[i].value* denotes the value of a pixel *i* within the structure that is *increased* by the boosting.

```
Data: Current structure s
Result: Visibility of s
for Pixel p in borderPixels(s) do
    for Pixel n in neighbors(p,!s) do
        if colorDistance(p,n) < JND then
            visible = false;
            break;
        end
    end
    cntVisible = visible?cntVisible+1:cntVisible;
    cntInvisible = !visible?cntInvisible+1:cntInvisible;
end
return cntVisible / (cntVisible+cntInvisible) > 0.99;
```

Algorithm 2: Tests the structure's visibility. The function *neighbors(p,!s)* provides the neighboring pixels of p within the 8 neighborhood that are not in the same structure.

5.3.3 Local Edge Preserving Color Mapping

While the color boosting algorithm of the last section is a supervised algorithm that requires the input of known visual structures that should be boosted, we present an unsupervised algorithm inspired by the local tone mapping algorithm of image generation for mosaic images of Meylan et al. (2007). This technique locally adapts each pixel by a model of retinal processing according to the non-linearities of the human visual system.

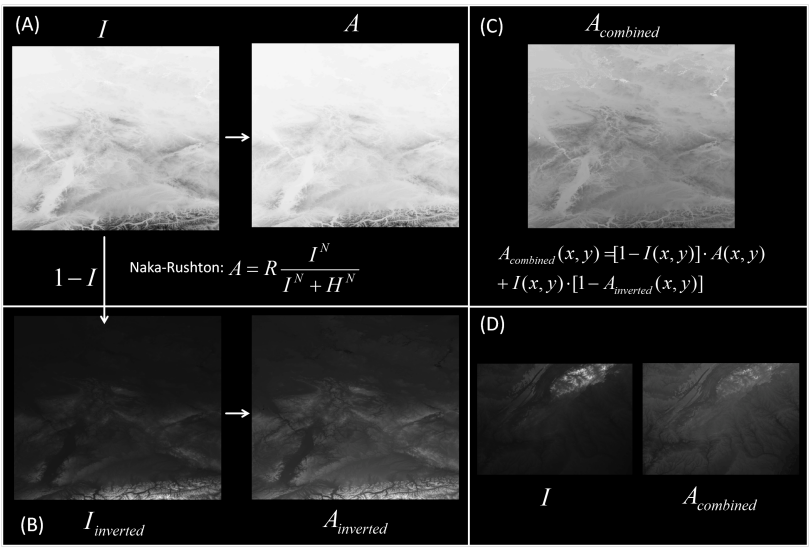


Figure 5.9: *Illustration of our edge preserving color mapping. (A) The data values of image I are convolved with a Gaussian low-pass filter in order to determine H, the intensity of the surround. The Naka-Rushton equation (Eq. 5.7) is applied to each pixel which results in A. (B) The same approach is applied to the inverted image $1 - I$ to make low contrasts and structures visible in the inverted result $A_{inverted}$. (C) Eq. 5.11 inverts the result of (B) and combines it with the result of (A). (D) The technique applied to a dark image example of topographic data.*

The idea is that our visual system compares the signal of one image pixel to its surrounding and adapts the perceived physical quantity of a signal to a perceptual stimulus. The non-linearities of the perceptual processes can be modeled by an equation by Naka and Rushton (1966) or Enroth-Cugell and Shapley (1973).

Meylan et al. (2007) adapt the Naka-Rushton equation for image processing in order to enhance the visibility of low contrasts in images (with $A(x, y)$ being the adapted intensity of pixel (x, y), the intensity $I(x, y)$ at pixel (x, y), the maximum intensity in the image I_{max}, the intensity of the surround $H(x, y)$, the maximum attainable signal $R(x, y)$, and the steepness N of the equation):

$$A(x,y) = R(x,y) \cdot \frac{I(x,y)^N}{H(x,y)^N + I(x,y)^N}, \text{ with } A, I, H \in [0,1] \tag{5.7}$$

$$R(x,y) = I_{max}^N + H(x,y)^N \tag{5.8}$$

The Naka-Rushton equation (Eq. 5.7) is a type of sigmoidial function that maps the ratio of the pixel intensity $I(x, y)$ to its surrounding $H(x, y)$ and thereby to a value from 0 to 1. Low contrast (slightly brighter pixels) in dark areas are mapped to larger value ranges, whereas high contrasts are mapped linearly (Meylan et al., 2007). The steepness of the function is controlled by N. If N increases, small difference in intensity will be assigned to higher values, which can be used to adjust the sensitivity of the technique. $R(x, y)$ is used to normalize the output. The intensity of the surround H is modeled by a Gaussian low-pass filter (with H being the input image I convolved with G, the normalization factor α, the Gaussian low-pass filter G, and the width σ of G):

$$H = I * G + \alpha \tag{5.9}$$

$$G(i,j) = \frac{1}{2\pi\sigma^2} e^{-0.5\frac{i^2+j^2}{\sigma^2}} \text{ with } i, j \in [-2\sigma, 2\sigma] \tag{5.10}$$

The parameters α and N are image dependent parameters but can be adjusted automatically (see Section 5.3.3). Given these parameters, the algorithm determines I_{max}, which is the maximum intensity of image I. Then it convolves I with the Gaussian filter and stores the result in H. The algorithm then applies the Naka-Rushton equation to each pixel and returns the result image A, which can be directly visualized (Figure 5.9 (A)).

This technique produces remarkable results for mosaic images, however, we

discovered two drawbacks when using it for data visualizations. First, the technique works very well to enhance the visibility in dark ranges of the image. However, it does not produce convincing results for brighter ranges (as in Figure 5.9 (A)). The reason is that the function maps low contrasts in bright areas into the upper asymptote of the function; thus, the adaption effect is reduced. Second, the parameter settings are sensitive to changes and cannot be easily adjusted for each visualization, especially for interactive techniques.

In order to enhance the local adaption for small differences in bright areas of images, we extend the existing technique of Meylan et al. (2007). The basic idea is to invert the image (Figure 5.9 (B)) and to apply the Naka-Rushton equation. In this way, also small differences in former bright areas are adapted. In order to combine both effects for dark and bright pixels, the results are combined by taking the original intensity of a pixel into account (Eq. 5.11). In this way, low contrasts in dark and bright areas profit both of a high adaption effect, whereas the high contrasts are preserved and thus, the global color distortion is kept at a minimum.

$$A_{\text{combined}}(x,y) = (1 - I(x,y)) \cdot A(x,y) + I(x,y) \cdot A_{\text{inverted}}(x,y) \tag{5.11}$$

$A(x,y)_{\text{inverted}} :=$ Adapted intensity of pixel (x,y) of inverted image I

Integration into Data Visualizations

This method can also be easily integrated into every pixel-based visualization. Therefore, the two dimensional image must be transfered into a two dimensional matrix containing the according data value for each pixel. This matrix can be directly used as input I in the Naka-Rushton equation, at which the data values replace the intensity values. During the convolving step in the algorithm, back-

ground, borders or other elements should be replaced by the local mean data value at the according coordinates. Since the algorithm works independently of the color map, other schemes different from grayscales can also be easily used. However, the color maps should be perceptually uniform and linear increasing in intensity (such as the heatmap in Figure 5.1) in order to avoid additional bias by a non-linear mappings and to fulfill the requirements of the analysis tasks (Section 2.3.1 (p. 44)).

Automatic parameter estimation

Meylan et al. (2007) used the following standard parameters: $\sigma = 3$, $N = 1$ and α was set to the half of the mean value of pixel intensity in the image. Besides the insensitive parameter σ (experiments are provided by Meylan et al. (2007)), the parameters N and α significantly influence the slope of the function (Eq. 5.7). Changes of α have the strongest influence on the result, since α increases the influence of the surround of pixels globally. Values closer to zero increase the adaption of lower contrasts but increase the shift of high and low values into the mid-range (gray-out effect). Higher values decrease the gray-out effect but also the local adaption. N controls the steepness of the function locally at each pixel. Thus, the visibility of low-contrast structures profit of higher N values (2-10). However, too high values will result in a binary edge image. In summary, in order to enhance the visibility of local structures, α should be as close to zero as possible and N should be as high as possible. However, in order to keep the color distortion at a minimum, α should be as high as possible and N should be as low as possible.

We, therefore, suggest an optimization algorithm that identifies the parameter setting that introduces a color distortion lower than a user defined threshold d (e.g. $d = 0.1$, which corresponds to a global color distortion of 10%). We apply our *local edge preserving color mapping* with varying parameters (α from 0 to 1 and e.g., N from 1 to 10). After each rendering the global distortion is computed with (Eq. 5.12). After all parameter settings have been performed, the algorithm

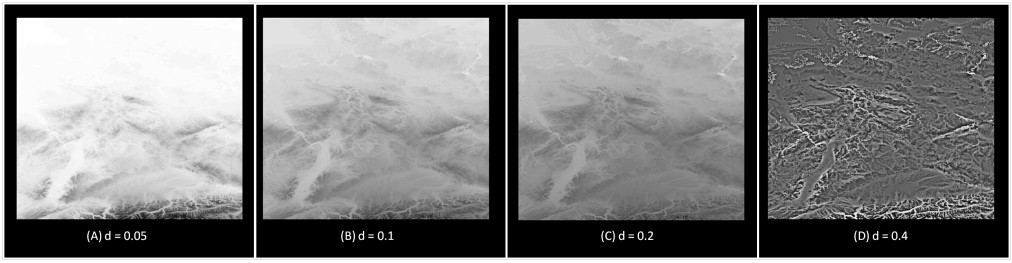


Figure 5.10: *Local edge preserving color mapping with different parameter settings. d is the user defined threshold that allows the maximum global color distortion ($d = 0.4$ corresponds to 40% distortion.).*

returns the result with a distortion equal or lower to the user defined threshold d. Different parameter settings are illustrated in Figure 5.10.

5.3.4 Evaluation

In this section, we compare our *local edge preserving color mapping*, our *color boosting algorithm* with two configurations, and the histogram-based color mapping of Bertini et al. (2007). As a baseline, we use a linear color mapping. We compare our color mappings to the histogram equalization of Bertini et al. (2007), because it is the only mapping, we are aware of, that can automatically map value ranges to uniform perceptual-distant color tones. In that way, it optimizes the perceptual impression of an image without user interaction or domain knowledge. The boosting techniques introduced by Oelke et al. (2011) are not considered for evaluation, because they are designed for sparse data distributions whereas our techniques address the problems of dense data distributions.

To compare the color mappings, we use two data sets: Smart Grid power consumption data (Figure 5.14 (p. 214) and Figure 5.15 (p. 216)) and topographic height map data (Figure 5.16 (p. 217)), which are presented in Section 5.4 (p. 213). The power consumption data is a typical example of a pixel based data visualization whereas the height map is based on fractal data. The evaluation compares the color distortion introduced by the evaluated color map-

ping approaches, the preservation of structures in the images, and the improved visibility of important points. Our color boosting algorithm requires an interest function that defines interesting points, which should be made visible. In order to evaluate the influence of different methods to determine interesting points, two different types (application and image based interesting points) are used in the comparison. Only for the Smart Grid data set, application-dependent interest points are available, whereas for the height map data set this information is missing. The image based interest function uses edge detection with the Sobel operator (Figure 5.7 (p. 196)) and defines the identified edges as important. The other important parameter for our method is the JND threshold. We select a $\mathrm{JND} = 1$, because according to the definition of the DIN99 color space a distance of 1 corresponds to a just noticeable difference for an average user. For local edge preserving color mapping, we allowed a maximum global distortion of $d = 0.2$ (20%). This setting was selected, since the competitive histogram-based approach also induces approximately this amount of distortion.

Measures

The compared techniques distort the color mapping locally to improve the visibility of interesting points. For the comparison, we are, therefore, interested in (1) the amount of color distortion introduced by the methods, (2) whether the interesting points are better visible than before, and (3) how well the structures in the baseline image are preserved by the techniques.

Color distortion.

To measure the amount of color distortion of a technique, the resulting image is compared to the baseline image created with the linear color mapping. The color distortion CD is then calculated as the sum of color difference of pixel colors in DIN99 color space (see Section 1.6.1 (p. 24)); with w, h the width and the height of the images. m denotes the image created with the selected

technique and l the baseline image created with the linear color mapping:

$$\mathrm{CD}_m = \frac{1}{w \cdot h} \sum_{x=1}^{w} \sum_{y=1}^{h} \Delta E \left(m(x,y), l(x,y) \right) \tag{5.12}$$

Visibility of Important Structures.

The second measure evaluates the visibility of important pixels that are defined for the specific applications. We cannot measure the visibility of the pixels directly, because an important pixel might be surrounded from pixels with the same color. In this case, all techniques will also change the color of the surrounding pixels in order to preserve the local color impression. To address this issue, we define a connected area of the same color as a structure and a structure containing an important pixel as an important structure. To measure the visibility of an important structure Algorithm 2 (p. 198) is used, which defines a structure as visible if 99 % of its border pixels are visible. We then calculate the percentage of visible important structures as the measure how well the technique is able to improve the visibility of important pixels.

Structure Preservation.

In addition to color distortion and the visibility of important pixels, the preservation of the structures in the original image is an additional interesting property. To represent the structures of the image, we use edges found with the Sobel operator in the ground truth image. In order to measure the structure preservation of a technique, the edges in the resulting image are compared to the edges of the ground truth image and the recall is calculated. The recall of edge pixels measures how many of the original structures are preserved by the technique. We do not calculate precision, because all techniques improve the visibility of barely visible structures and make additional structures visible, which in turn will result in very low precision values.

Visible Edges.

The structure preservation measure reveals how many edges of the baseline image are preserved in the non-linear mappings. However, these edges might not be visible to a human analyst. However, since edges (variations in the data) carry valuable information, another interesting question is, how many of these edges are perceivable by an analyst after performing each technique. Therefore, we measure at each edge the color distance to its neighbors. If the distance is above the just noticeable difference (Section 5.3.1 (p. 189)), we consider the edge as visible. The recall will reveal, how many of the edges do actually contribute to visual analysis after the different mappings. Again, we do not measure precision due to same reasons as named above.

Results & Discussion

Table 5.1 and Table 5.2 show the evaluation results for the Smart Grid data set with the rectangular and spiral visualizations. The results for the topographic height map data is shown in Table 5.3.

Looking at the results on the Smart Grid data set with the rectangular visualization in Table 5.1 and comparing the results of the color mapping techniques with the baseline, it gets clear that all techniques improve the visibility of application dependent interest points. Beside the color boosting technique with the application specific important points, also all the other techniques improve the visibility of edges. The histogram equalization technique is harming some edges in the original image and introducing the most color distortion. The other techniques preserve the structures in the image and the color boosting variants are introducing only a little color distortion.

The results on the spiral visualization in Table 5.2 are similar except for the histogram equalization technique. This time, the technique preserves the structure much better but does not improve the visibility of edges as much as the other methods.

Table 5.1: *Evaluation results on the rectangular pixel visualization (Figure 5.14 (p. 214)).*

	Visible Important Pixels [%]	Visible Edges [%]	Preserved Structures [%]	Color Distortion [%]
Linear (baseline)	26.59	63.21	100.00	0.00
Bertini et al. (2007)	49.60	90.64	85.94	21.12
Local edge preserving color mapping	61.11	98.36	97.60	13.28
Color Boosting (edges)	45.24	100.00	99.98	0.78
Color Boosting (application)	100.00	68.81	99.51	1.28

Table 5.2: *Evaluation results on the spiral pixel visualization (Figure 5.15 (p. 216)).*

	Visible Important Pixels [%]	Visible Edges [%]	Preserved Structures [%]	Color Distortion [%]
Linear (baseline)	39.41	68.13	100.00	0.00
Bertini et al. (2007)	46.61	83.17	91.28	25.82
Local edge preserving color mapping	72.03	96.36	98.74	18.93
Color Boosting (edges)	43.22	90.19	99.68	1.04
Color Boosting (application)	96.19	69.60	98.43	5.86

Table 5.3: *Evaluation results on the topological map data (Figure 5.16 (p. 217)).*

	Visible Edges [%]	Preserved Structures [%]	Color Distortion [%]
Linear (baseline)	47.00	100.00	0.00
Bertini et al. (2007)	88.84	93.78	43.35
Local edge preserving color mapping	69.65	90.64	17.37
Color Boosting (edges)	81.50	95.34	3.52

For the results of the topological height map data, shown in Table 5.3, the situation is different. The histogram equalization technique is better in making the edges in the image visible. For this image, all techniques are losing some structure information during the color mapping. For the color distortion we get similar results than before. The histogram equalization introduces the most distortion in color and the color boosting technique the least distortion.

All color mapping techniques improve the visibility of structures in images but introduce some distortion in color and losing some structure information. The color boosting technique is focusing on predefined important pixels. Consequently, in cases where important pixels can be determined, this technique outperforms the others. As it is only changing the important pixels, it introduces the least color distortion compared with the other techniques. The histogram equalization and the local edge preserving color mapping are independent from important points, therefore, it is no surprise that these methods are not gaining high improvement on predefined important points. Their advantage is the ability to improve the visibility of edges in the images. The major difference between these two techniques is the amount of color distortion, which is much higher for the histogram equalization than for the local edge preserving color mapping. Interesting is that the histogram equalization is much better in improving the edges on the map data than our techniques, which is not the case for the pixel visualizations.

The conclusion of the evaluation is that there is no clear winner. Every technique has its strength and weaknesses. The color boosting technique introduces the least color distortion and in case important pixels are known, this technique is best to improve their visibility. Comparing the local edge preserving color mapping with the histogram equalization, the local edge preserving color mapping introduces less color distortion but if fractal images are used, the histogram equalization is better suited to improve the visibility of structures in the image. The disadvantage of the histogram equalization technique is the huge color distortion, which is intended by the method but makes this technique unusable if users should be able to compare color values.

5.3.5 Heuristics for Contrast Enhancement

The techniques presented above aim to maximize the visibility of visual structures by minimal distortion of the color encoding. Thereby, they aim to support the task combination of *identification* and *localization* of data values *faithfully*. In this section, we provide efficient heuristics and best practices maximizing the readability of important data values in high-frequency or dense visualizations. Therefore, these heuristics only aim for supporting the *localization* task.

Boosting with Visual Attention.

The most prominent approach to highlight visual elements with color is to attract visual attention. Section 2.3.3 (p. 51) shows that humans are predominantly attracted by bright and saturated colors. For accurately reading color encoded data it is required that colormaps contain many distinct color hues with equalized saturation. Therefore, intensity of colors can be used to highlight (boost) important data values (combining elementary identification and localization). In order to highlight some important data structures, we reduce the intensity of other visual structures (we set intensity to 50% of the original color in the HSI color space (Keim, 2000)). Thereby, important visual structures with full intensity are visually boosted and the analyst can locate them on the screen. At the same time, the analyst is able to read the values from all visual structures since saturation and hues of visual structures are not adapted, which allows effective identification of visual structures (Section 2.3.3 (p. 51)). Figure 5.11 shows a pixel-based visualization visualizing a time series of power consumption measurements that is layouted in recursive patterns (Keim et al., 1995). Each pixel block is on hour, 24 blocks form one day, days are horizontally layouted, and weeks vertically. The colors are fully saturated but at 50% of intensity and encode the power consumption. Intensity is used to encode the anomaly score, which indicates important values for the analysis of large scale power consumption. Thereby, the color mapping highlights time frames in which the measured power consumption was not expected guiding the analyst towards in-

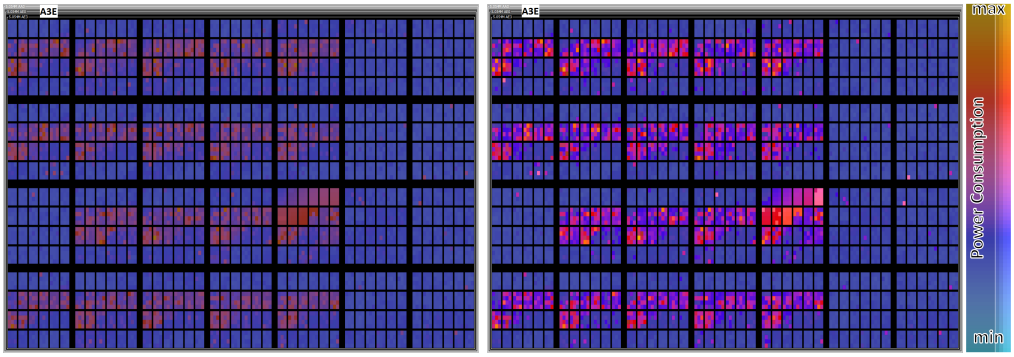


Figure 5.11: Image taken from [7]. *On the left, power consumption is visualized without visual boosting. On the right, the intensity of colors visually boosts some anomalous data values that were not expected guiding the analyst towards important time frames in the overview.*

teresting time frames in the overview.

Boosting with Color Contrasts.

We experienced that it is hard for the user to perceive single visual structures, if many visual structures occur in dense areas of the display. Studies have shown that the visibility of low contrasts is reduced in high spatial frequency areas of the display (Barten, 1999) due to contrast sensitivity. The method presented in Chapter 4 uses a perception model to estimate the bias of contrast effects and contrast sensitivity. The method is able to compensate for the loss of high-frequent information by increasing the contrast of high-frequent visual structures. This method has, however, two disadvantages: First, the method only considers the bias of perception and not the bias of color encodings. Second, the method is computationally expensive.

Therefore, we provide a heuristic to visually boost visual structures with contrast effects in displays with high spatial frequency. With the perception model of Chapter 4, it is possible to approximate a color c' that has a maximum color contrast to a target color c with $c'_x = D65_x - c_x$ (with x being the LMS channel in the CAT02 color space and $D65$ is the standardized reference light). Note, that this is only valid for saturated colors where one of the channels is close to zero. In order to visually boost the readability of visual structures in visual-

izations, borders can be drawn around the visual structures. These borders are encolored with maximum color contrast to the segment color.

This accords to *boosting with color* by Oelke et al. (2011), which adapts the surround of pixels by less saturated colors of the same hue, which increases the readability by the "chroma chrispening" contrast effect. This effect, however, requires more space of the surround since our eye is less sensitive to spatial frequency in the saturation channel (see Ware (2012) and Section 2.5 (p. 99)). The spatial sensitivity of the lightness channel is far higher. In order to minimize the required area around the important pixel or structure, we boost the color by contrast effects in lightness and hue.

In order to increase the contrast effect, the intensity should be adapted according to the background color. If the background is dark, the intensity of the border colors should be maximized (1.0 in the HSI color space (Keim, 2000)). Else, the intensity of the border colors should be minimized such that the color is still clearly visible to preserve the color contrast to the visual structure (0.3 in the HSI color space (Keim, 2000)).

It is recommended to enable the user to control the segment and border sizes, which increases the effects and readability. On the one hand, these contrast effects can bias the user in reading color encoded features if the task is a detailed analysis of visual structures. On the other hand, this approach enhances the perception and recognition of visual structures in overviews, which enables the analyst to read features even for dense areas on the screen. Therefore, we recommend the approach for tasks that require overviews with high frequency patterns. For detailed analysis, where the focus is on reading features from specific (zoomed-in) data objects, the boosting should be avoided. If user interaction is not desired, the optimal width of visual structures and their borders can be estimated by the contrast sensitivity function (Section 4.4.2 (p. 162)). The human eye detects patterns with 1.5 to 7 cycles per degree of the visual angle with a peak sensitivity at 3-4 cycles per degree. Therefore, the width of visual structures and their borders should not be smaller than 7^{-1} degrees of the visual angle. According to the viewing distance D, display width N, and resolution n, the width in pixels can be calculated with Eq. 4.6 (p. 164). For $D = 61cm$,

$N = 31cm$, and $n = 1920px$ (horizontal), the width of the structure should be at least $4px$ and the borders $4px$ and at best $10 + 10px$.

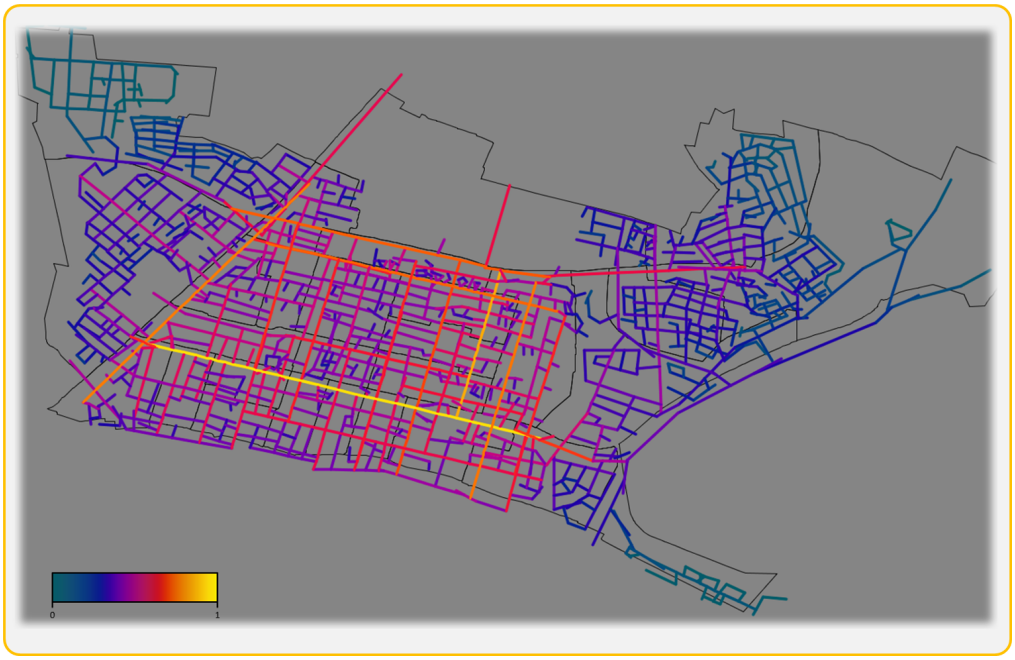


Figure 5.12: Image taken from [18]. *Overview of a street network. Color encodes different features for exploratory analysis (here: centrality of streets).*

This method requires additional screen space for the borders. Therefore, it is well applicable in applications where visual structures are spatially separated, e.g., street-network analysis; or in applications in which visual structures, that are not important, can be overdrawn to boost the visibility of important visual structures. Figure 5.12, shows an application of street-network analysis. Streets are visualized on a map with color encoding different attributes such as length, connectivity (number of connected streets), or centrality (graph-based betweenness measure) of a street in a city. The application supports similarity searches such that the user is able to select a street. Streets with similar features should be highlighted. The overview visualization has typically a high spatial frequency and it is not easy for the human eye to detect every street. Therefore, we apply both heuristics *boosting with visual attention* and *boosting with contrast effects* to support the highlighting task. Figure 5.13(a), shows a region without

boosting. In (b) some streets are boosted with visual attention, which is further increased by adding borders that cause strong contrast effects to fore- and background (c). In (d), the width of streets and borders is adapted to exploit higher contrast sensitivity of the human eye.

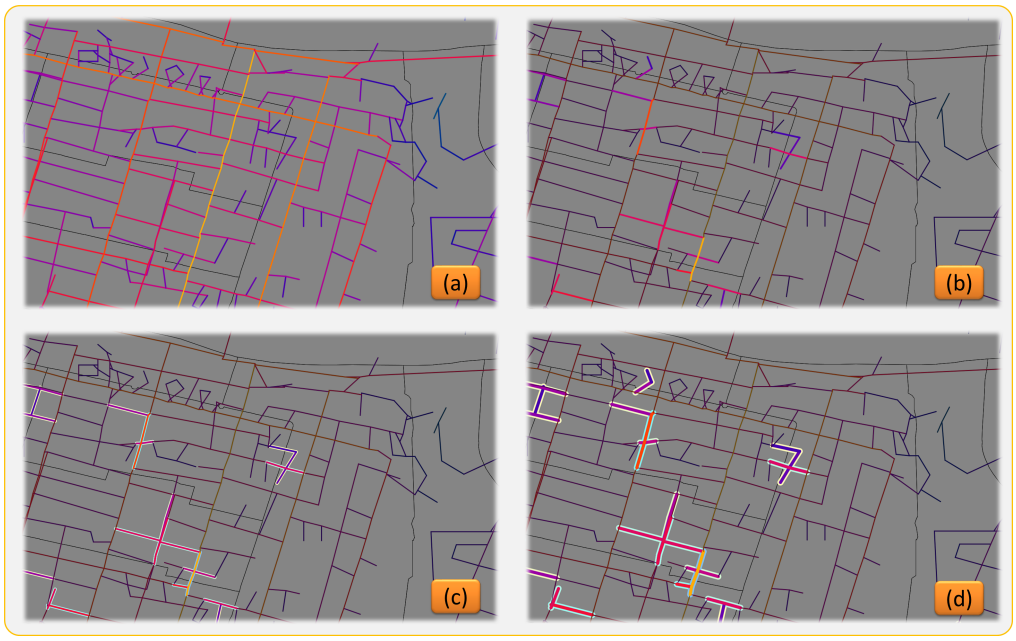


Figure 5.13: Image taken from [18]. *Boosting of a high-frequency street-network visualization. The analyst searches for street segments with similar features to a target street, which are boosted by the visualization. (a) No boosting. (b) Boosting with visual attention. (c) Boosting with visual attention and contrast effects. (d) Adapting the width of streets and borders to increase contrast effects exploiting the contrast sensitivity of the human eye.*

5.4 Use Cases

In this section, we apply our techniques to several pixel-based visualizations, including recursive patterns, spiral visualizations, and scatter plots, demonstrating their effectiveness and applicability.

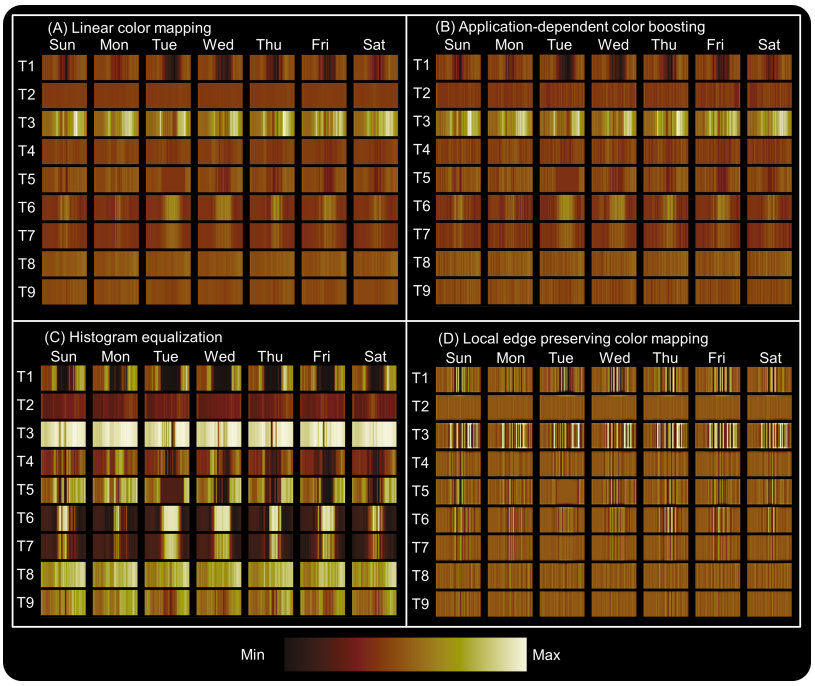


Figure 5.14: *Boosting of important points in pixel-based time series visualization. This image illustrates a pixel-based visualization of energy data with different color mapping techniques. The data from nine transformer stations (T1-T9) is shown for a time period of seven days. Each rectangle holds 24 hours of data from left to the right. (A) Linear color mapping; (B) Our application dependent color boosting; (C) Histogram-based technique Bertini et al. (2007); (D) Our local edge preserving color mapping.*

5.4.1 Smart Grid Management

One motivation of Smart Grids is their ability to include regenerative energies such as solar or wind energy. The Smart Grids are required, because the amount of power produced with regenerative energy cannot be predicted reliably. A Smart Grid observes the status of the energy grid and the operators of the grid are able to react on sudden changes in the production or consumption of power.

A typical question in a Smart Grid environment is the analysis of past behavior to derive actions for the future to improve the grid stability. An analytical problem in this case, is the analysis of power transported through transformer

stations of a local grid. A pixel-based visualization is used in Figure 5.14 to compare several transformer stations in parallel. To analyze regular daily trends, the spiral visualization in Figure 5.15 is used. A problem in both cases is the comparison of different transformer stations. Depending on consumers and producers connected to a transformer, the local patterns are different. Some transformers have very high variance in the data whereas others have only small variations.

Figure 5.14A shows this problem of a linear color map. In this case, the consumed energy on nine transformer stations is shown with a linear color mapping, which is normalized between the extreme values of all transformer stations. The large range of values in combination with a linear mapping leads to the situation that small variations in the time series are not visible. For instance, the variations of time series T2 are not visible in this case. But still these variations are important for the analysis. To solve this problem, the presented techniques are used with an application specific interest function that detects interesting changes of the energy consumption. The results (Figure 5.14B) of the color boosting reveals these small variations in the data but the global color impression is still similar to the original mapping. In contrast, the local edge preserving color mapping is boosting every variation, which results in an image that stresses all changes in the time series (Figure 5.14D). The local edge preserving color mapping, thereby, introduces two artifacts. First, certain values are moved to a completely different value range, for instance, in T1 or T3. Second, the local edge preserving color mapping increases the difference between the border and the content in some situations, which is visible for time series T3. Thus, vertical pixel lines, representing one time interval of the data, do not have the same color over their whole length. Using the histogram equalization (Bertini et al., 2007) (Figure 5.14C) many structures are visible but the disadvantage of this technique is the non-linear distortion of the color map which destroys the perceptional linearity of the color map.

A similar observation can be made with the different color mapping techniques and the spiral visualization in Figure 5.15. The linear color mapping is not able to show the small variations of values due to high outlier values.

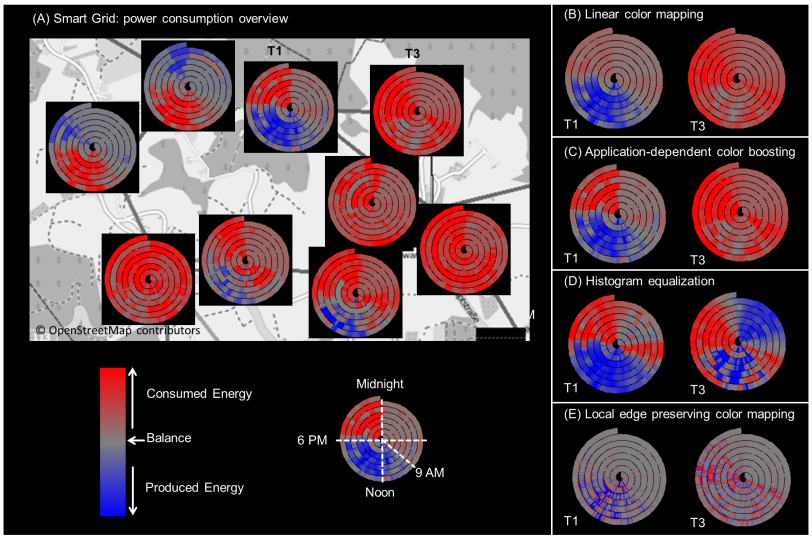


Figure 5.15: *(A) Seasonal trend visualization for energy production and consumption on a map. One week of energy data is visualized by time series spirals, with increasing time from the inner to outer ring. Blue indicates net power production (surplus), whereas red reveals net power consumption. (B) Linear color mapping; (C) Our application dependent color boosting; (D) Histogram-based technique Bertini et al. (2007); (E) Our local edge preserving color mapping.*

The histogram equalization technique is able to highlight these variations but the bipolar color mapping is harmed. Only the color boosting technique with an application specific interest function is able to show the local daily peaks but preserves the meaning of the bipolar color map. The local edge preserving color mapping is highlighting the edges in the image and harms, thereby, the color impression.

5.4.2 Topographic Height Map

Another application of local adaptive color mapping techniques are topographic height maps (see Figure 5.16 for a height map of Germany). Because of the unequal distribution of height information, the linear mapping is not able to show

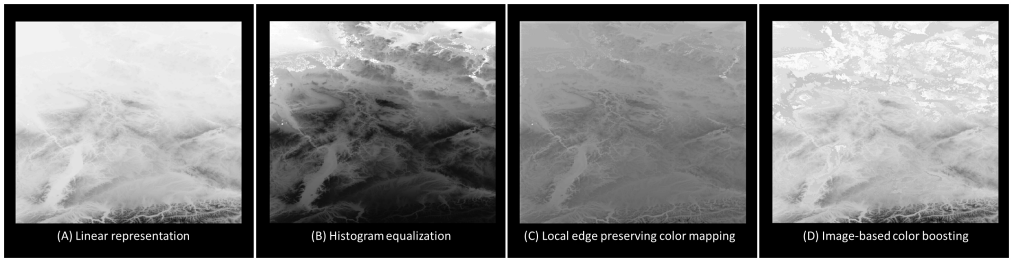


Figure 5.16: *Topographic data (x-axis: longitude; y-axis: latitude; intensity: height), using linear min-max normalization for color mapping (A), a histogram-based technique (Bertini et al., 2007) (B), our local edge preserving color mapping (C) and our image-based color boosting (D). The local structures in the bright areas of (A) remain hidden, that are revealed in (B), (C) and (D). Structures of the darker areas are removed by the histogram-based approach and the global comparability is biased by the non-linear color scale. Our techniques reveal both, the structures in the bright and dark areas, and keep the global distortion at a minimum. In particular, the color boosting boosts all important image structures above the just noticeable difference and guarantees their visibility at the cost of biasing the continuity and natural forms within the data. These are preserved by the local edge preserving color mapping, which introduces more distortion than the color boosting but less than the histogram approach.*

details in the brighter areas (A). As we lack an interest function for this data set, we decided to use for the color boosting the image-based detector introduced in Section 5.3.2 (p. 195) to identify important structures in the image. The color boosting, therefore, highlights the edges especially in the northern part and the visibility of structures is enhanced, compared to the linear mapping. But the boosting is by far too strong and results in a much too hilly impression (D). In this case the local edge preserving color mapping results in a better image (C). The structures are much better visible than with the linear mapping and relations of heights are much better preserved than with the color boosting technique. We can also observe the property of the local edge preserving color mapping to map colors near the extreme values to the middle of the colormap, which results in a more grayish image. Comparing the results of the histogram equalization (B) with local edge preserving color mapping shows that the structures in the north are better visible with the histogram technique. However, the histogram technique is not able to show the fine structures in the southern part of the map.

5.5 Discussion & Future Work

We present two automatic techniques that locally adapt color mappings to improve the visibility of interesting pixels and structural information in order to enhance the *expressiveness* of color encodings by supporting the *identification* and *localization* analysis tasks. The local edge preserving color mapping is an unsupervised algorithm whereas the color boosting technique requires predefined important pixels. Both algorithms are applicable to any kind of 2D image or visualization.

The evaluation shows that both techniques are introducing little color distortion and are applicable to dense pixel based data visualization techniques. In case important pixels are known, the color boosting technique outperforms the state-of-the-art. For example, in data visualizations where an effective global color scheme hides 75% of the local data patterns, the supervised boosting method reveals at least 96% of the local data patterns with a color distortion of only 1.28%. The state-of-the-art only reveals 47% with a distortion of 25%.

General Limitation

It should be highlighted that there are some general limitations in the local-adaptive color mapping methodology and there are interesting future work items to be addressed. When local adaption is applied, artifacts might be introduced, since equal data values may be mapped to different color tones and different data values may be mapped to equal colors, based on the local adaption strategy. In these cases, the analyst is biased in the comparison task. This limits the *effectiveness* of color encodings by decreasing the *faithfulness* of the encoding.

However, it is an interesting fundamental question how we can assess the trade-off between the analytical gain of making local structures visible and the bias, which is introduced by this local adaptive methodology. On the one side, the visibility of local structures increases the *expressiveness* and supports *local-*

ization and *identification* tasks. On the other side, the introduced bias decreases the *faithfulness* in the *comparison* tasks. Research towards a more detailed understanding of the trade-offs for different application domains and visualization techniques is, therefore, an interesting challenge.

When to use the techniques?

The general challenge of the local-adaptive color mapping methodology is to enhance the visibility of local differences while keeping the global distortion at a minimum. We addressed the general limitation by optimizing the distortion globally in local edge preserving color mapping and locally in our color boosting technique based on the just noticeable differences (JNDs). However, artifacts are produced in the color boosting technique, if one part of the surrounding of a structure is lower and the other part is higher. In this case, the structure will be boosted either higher or lower than both in the final image. The structural information is then visible, however, the continuity and natural forms within the data are biased.

This leads to the conclusion that the color boosting technique should rather be used in visualizations of metric and abstract data than in visualizations of natural forms and structures (Section 5.2). The presented local edge preserving color mapping preserves the natural form within the data, but tends to scale low contrasts into the middle of the function, which would lead to a gray-out effect, if very fine structures have to become visible. This leads to the conclusion that this method should rather be used in visualizations of natural forms and structures.

When to use the heuristics?

The heuristics are efficient means and best practices for boosting information in visualizations by color. While the techniques above aim for the combination of *localizing* and *identifying* data values (thereby, aiming for *faithful* color encodings), the heuristics aim only to maximize support for the *localization* task.

The heuristic *boosting with visual attention* also maps equal data values to different colors and can bias the user in the *comparison* task. It uses light-

ness for boosting colors, which is the most effective channel for the human eye to perceive magnitudes. Therefore, the analyst pre-attentively perceives the boosted data values with higher magnitudes than the regular data value, although it shares the same value, same hue, and same saturation. However, it is applicable the dense data visualizations and highly scalable. Similar to the color boosting technique, this heuristic requires the application dependent information of important visual structures for the boosting. It maximizes the lightness for boosting, while the color boosting technique makes visual structures "just visible". Therefore, this heuristic should rather be used in localization tasks only.

Boosting with color contrasts does not change the color of the data value but the color of its surrounding. Therefore, this method requires visualization space around encoded data values and cannot be used in dense visualizations. The method maximizes the contrast for significantly increasing readability in sparse or high-frequency visualizations (e.g., street networks). If enough background pixels around visual structures are available, this heuristic should be used for boosting.

Interactive Pattern Boosting

We see interesting challenges in researching and extending image-based structure detectors for visual analytics applications. It would be interesting to develop an interactive approach where the user marks a local data pattern of interest, and then the systems computes a color mapping which best emphasizes this local structure in the global view. Thereby, local-adaptive color mapping could be conveniently parameterized on-the-fly in an application-dependent way.

Combination with Contrast Effect Compensation

The local-adaptive color mapping methodology should be used before the methods of contrast effect compensation of Chapter 3 and Chapter 4. The methods of Chapter 3 and Chapter 4 can only preserve effective color perception and the visibility of structures of the input image. If the color encoding maps structures in a way that they are not visible from their surround, the method of contrast

effect compensation will intentionally preserve this appearance, because this is considered *faithful* according to the mapping. If the methods of contrast effect compensation are applied after local-adaptive color mapping techniques, they will, however, preserve the visibility of the boosted structures since these are then included in the rendering. Further, the contrast effect compensation methods will also boost structures that did not become visible by the boosting techniques due to perceptual issues such as contrast sensitivity.

Thus, the combination of both techniques — boosting and contrast effect compensation — enhance the *faithfulness* and *expressiveness* of *effective* color encodings.

6

Concluding Remarks and Perspectives

"Effective color encodings are pre-attentive, semantic consistent, faithful, expressive, and support the target analysis task."

This thesis describes and tackles the challenges of encoding information effectively with color. The main contributions are: 1) a novel definition and novel quality metrics to describe and measure the effectiveness of color encodings; 2) novel guidelines and methods to effectively encode information with color for (combined) analysis tasks; (3) the first methodology and method that compensates for physiological biases, which provides a solution for the problem of contrast effects in information visualization; (4) and an extension, which captures individual perception of contrast effects and can be personalized to the individual target users; (5) novel methods that preserve the effectiveness of color encodings by boosting information with color.

For the most applications of visualizations, the question of "how to encode information with color effectively?" can be answered with a combination of the methods and guidelines in this thesis (exceptions can be found below). The recommended work-flow is:

1. Design an effective color encodings for the target analysis tasks with the guidelines or the tool *ColorCAT* of Chapter 2 to satisfy the requirements of *effective* color encodings.

2. If, and only if, there are important data patterns that must be visible in the final visualization or to increase the readability of high-frequency visualizations, the *expressiveness* of the color encoding can be increased by integrating the methods of Chapter 5 as a post-processing step (before contrast effect compensation).

3. Configure the perception models for contrast effect compensation to the individual target user and system with the methods of Chapter 4.

4. Apply the methods of Chapter 3 or Chapter 4 to compensate for contrast effects in order to provide *faithful* color encodings adapted to the individual user.

This thesis presents with diverse use cases, applications, and experiments that the introduced methods provide effective color encodings. With the provided guidelines, visualization researchers and designers can design effective colormaps for their target application, target domain, and target users.

The methods for contrast effect compensation can be applied as post-processing step on every image or data visualization. Therefore, the methods and methodologies can and should be used in visualizations that represent information with color. The experiments reveal that the accuracy of users reading and comparing color encoded data is significantly improved (the error decreases from 24% to 10%) in comparison to (state-of-the-art) color mapping. Exceptions are localization tasks and visualizations of streams, natural forms, and shapes in which the perception of contrast effects is required for effective detection of edges and highlighted data points.

The next section will summarize the contributions of this thesis in more detail and the last section provides perspectives and open research challenges for future research.

6.1 Summary of Contributions

Chapter 2 reveals the importance of the target analysis task for the effectiveness of color encodings. By the requirement analysis of the perceptual foundations and analysis tasks, it becomes clear why some colormaps perform well in some tasks and poor in other tasks. The state-of-the-art proposes guidelines for single elementary tasks. However, real applications, especially those of exploratory nature, require a combination of elementary analysis tasks. Therefore, Chapter 2 introduces novel quality metrics, guidelines, and methods to encode information — single dimensions and high-dimensional data relations — effectively with color; and, further, provides the tool *ColorCAT* that guides visualization experts through the design process of colormaps for combined analysis tasks. Thereby, the visualization expert is able to adapt and design colormaps for the application, preference, and culture of the target user.

The perception of color encoded data depends on the surround in the final rendering. Therefore, contrast effects cannot be overcome with the existing guidelines and rules-of-thumbs that aim to avoid contrast effects before the visualization is rendered. Chapter 3 provides a novel methodology and method based on novel quality metrics and existing perception models to measure and compensate for physiological biases. Thereby, the method provides a solution for the problem of contrast effects in information visualization. The results of two experiments with over 40 participants reveal that this method doubles the accuracy of users reading and comparing color encoded data values. Further, the method is application independent and can be used in any visualization system, which is demonstrated in various application examples.

Chapter 4 proposes an extension to existing perception models that allows personalization to capture individual differences of contrast effect perception. Further, this chapter provides novel methods to adapt contrast effect compensation to the individual user, to viewing distances and different environments (e.g.,

ambient light and display gamut). An experiment reveals, that the personalization of contrast effect compensation further reduces the error of users reading and comparing colors by 29% compared to the method that applies existing perception models.

Global color mappings often hide important local data patterns. Existing methods of data transformation and color mapping cannot reveal this local information without distorting linear encodings, which biases the user in comparing data values significantly. Chapter 5, therefore, provides novel methods that locally adapt the color mapping to boost local information in displays that is not visible to the user. This general idea of locally adapting colormaps has the challenge to visualize all important data structures (to increase expressiveness) and to minimize the distortion of the color mapping (to preserve the faithfulness of the color encoding) at the same time. In experiments with real applications, well-designed global color mappings hide 75% of the local data patterns. The novel local adaptive color mapping reveals 96% of the local data patterns with a color distortion of only 1.28% and, thereby, significantly outperforms the state-of-the-art, which only reveals 47% with a distortion of 25%.

6.2 Future Perspectives & Open Research Questions

Compensation of Optical Biases

Our compensation method is application independent and can be applied on any data visualization or image and is, thereby, a general method that should be used in data visualizations that encode information with color.

The effectiveness of our method depends on the perception model. We decided to apply the iCAM framework (Fairchild and Johnson, 2004) since it is

robust in predicting color appearance, simultaneous contrast, and chroma christening in images with local *von Kries* chromatic adaptation (von Kries, 1905). However, there exists a variety of physiological illusions that cannot be predicted. For example, the Howe- and White illusion requires a more complex consideration of the surround. Blakeslee et al. (2005) and Robinson et al. (2007) apply oriented difference-of-Gausian filters to predict a variety of simultaneous brightness effects, covering Howe- and White illusion. Similar to the iCAM framework, they are operating on the lowest levels of brightness perception.

Our method is formulated as a numerical optimization problem and solved by a gradient-based optimization approach. The perception model must, however, not be differentiable for our approximation of the gradient (see Section 3.3.3 (p.121)), but roughly *homogeneous* (contrast effects will shift the pixel color in the same way regardless of the initial pixel color) and *invertible* (contrast effects on one single pixel can be compensate in the inverted direction of the bias). Therefore, perception models based on linear *von Kries* chromatic adaptation (von Kries, 1905), Gaussian- or difference-of-Gaussian filters can be directly applied in our method, which covers a broad variety of models for the lowest levels of color perception such as the methods of Blakeslee et al. (2005) and Robinson et al. (2007). However, it is not clear how to combine these different models, since most of these models operate only on grayscale images, while iCAM does predict color appearance. Therefore, it would be beneficial to find means to combine models of brightness effects and color appearance.

Higher level models that require a decomposition of the image in different layers or complex sets of discrete rules are not supported by our method since the predicted effects cannot be inverted accurately to approximate the gradient of compensation. For example, Anderson and Winawer (2005) present a theory and evidence that perception of surface lightness decomposes the scene / image into different layers. The anchoring theory of Gilchrist et al. (1999) predicts lightness perception with a set of complex rules and successfully explains anomalies and systematic biases. It is an interesting and promising research challenge to find computational models for such theories that would allow the compensation of higher-level or even cognitive biases.

Design of Visual Variables

The effectiveness of color is due to its perceptual channels to perceive ordinal and quantitative as well as nominal information. These dimensions allow that color encodings can be effectively designed for different target analysis tasks and also their combinations. Other visual variables could also be designed such as shape, texture, and glyphs in general.

These variables are often used to encode categories. To encode ordinal and quantitative data, however, a perceptual kernel that measures similarity and dissimilarity of values as well as a measure of perceptual order is required. The dimensions of the perceptual kernel and the measures of perceptual order can form the dimensions of a perceptual uniform space for visual variables such as shape, texture, and glyphs. Thereby, it would be possible to find means to design visual variables with these dimensions, similar to the guidelines and methods in this thesis. Research could develop quality measures for effectiveness and guidelines to design encodings with the visual dimensions for the analysis tasks of target applications. Recently, Demiralp et al. (2014) estimated the perceptual kernel for shapes, which is the first step towards the direction of designing other visual variables.

Device Dependend Colormapping

Although there exist standards for displaying and printing colors, practice shows that colors always appear different on different devices. There are methods to correct color displays and to configure devices as well as ambient light (see Section 4.4.3 (p. 166)). Also the novel methods presented in Section 4.4.3 can adapt contrast effect compensation to target environment settings and aims to compensate for skewed color appearance. However, the output display is not considered in the initial design of the color encoding. There is a clear need to correct the appearance of colormaps and to preserve its properties on different displays. For example, perceptual linearity may be skewed if perceptual linear colormaps are designed on LCD displays and then printed. We see high potential in integrating the human in the loop to capture the individual characteristics

of the hardware setting but also to capture individual differences of color perception. Thereby, the initial color mapping could already become effective on the target display, which could then be preserved by the contrast effect compensation methods.

Color Blindness

The guidelines and methods for designing effective color encodings in this thesis consider color vision deficiency. Models that predict color blind vision are used in *ColorCAT* to reveal designers how a color blind person will perceive the color encoding. To verify the implementation of *ColorCAT*, color blind persons were shown different images in normal color vision mode (Figure 6.1 (a)) and in color blind vision mode (Figure 6.1 (b)). The participants could not determine any difference between the images, which approves that the simulation models accurately predict color blindness perception.

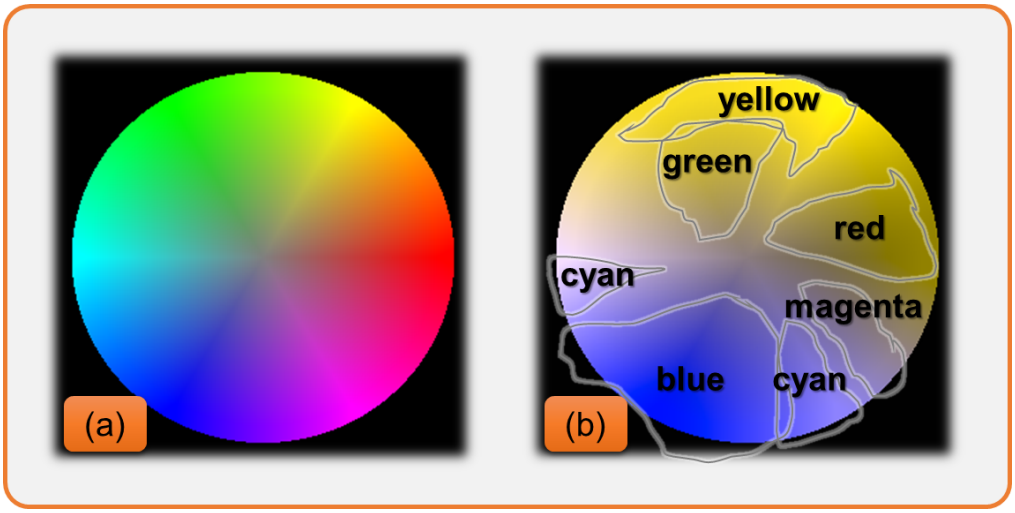


Figure 6.1: *Color hue perception of color blind persons. (a) Normal color vision. (b) Simulation of red-green blindness (Brettel et al., 1997). Color blind persons (deuteranopia) cannot determine a difference between (a) and (b). However, persons with normal color vision just see yellow and blue hues in (b), while color blind persons perceive saturated hues as indicated by one participant who draw the hue areas in (b). This indicates that color blind persons* ***learn*** *to perceive hues and that models of color blindness* ***are not sufficient*** *in predicting which color hues are perceived by color blind persons.*

However, when they were asked to point at the regions in Figure 6.1 (b) to determine specific hues, they were pointing at some areas. One participant draw the areas of hues on Figure 6.1 (b). This reveals an interesting finding: While normal color vision sees just yellow and blue, color blind persons can determine hues (Figure 6.1 (b)). While the prediction of color blindness works well in the sense that it correctly predicts how the colors are processed by color blind vision, it cannot predict how the persons perceive hue and, thus, cannot reveal to persons with normal color vision how the colors appear to color blind persons. The (color blind) perception of hues seems to be learned (the visual system learns to compensate for anomalous or missing physiology), we have to find models that can be personalized for accurate prediction of hue perception. Since the perception of hues is very important for the effectiveness of colormaps, we argue for more research in this area.

Color Anchors

During the experiments to evaluate the compensation of contrast effects, participants were interviewed how they performed the task of reading a color encoded value from displays supplied by a color legend. We observed, that participants tried to name the colors and many said that they interpolated the colors between two *color anchor* points, e.g., "I tried to find the color between yellow and orange". This leads to the question, "Could color encodings become more effective if there are many colors that can clearly be named, which act as anchor points to read colormaps?" There is research that showed that color perception and language interfere (Roberson and Davidoff, 2000). Lin et al. (2013) provided a method to estimate single categorical colors that are semantic resonant with certain topics. They found that semantic resonant colors are more effectively read than state-of-the-art colors for categorical data. However, their method is not applicable for quantitative and ordinal data.

We believe that integrating color anchors in colormaps has significant impact on reading and comparing color encoded data. Section 2.3.2 (p. 49) showed that the identification of quantitative data and absolute judgments are not yet *pre-attentive* because they require a color legend that is typically hard to learn.

There is high potential that the identification task with color could become *pre-attentive* if color legends are efficiently and effectively learned by remembering color anchors in a colormap.

ColorCAT could integrate color anchors by replacing the base colors in the colormap algorithm. However, there remain the questions: "How to identify color anchors? How to distribute color anchors over the colormap? How does this interfere with the perceptual quality metrics? And does this approach really have an impact?" Answers to these questions are challenging and hard to find but they build an interesting and novel future research direction.

List of Figures

My Publications

[1] Jürgen Bernard, Martin Steiger, Sebastian Mittelstädt, Simon Thum, Daniel A. Keim, and Jörn Kohlhammer. A survey and task-based quality assessment of static 2d color maps. In *Proceedings of SPIE 9397, Visualization and Data Analysis*, page 93970M, 2015. 20, 32, 43, 55, 93, 94

[2] Fabian Fischer, Florian Stoffel, Sebastian Mittelstädt, Tobias Schreck, and Daniel A. Keim. Using Visual Analytics to Support Decision Making to Solve the Kronos Incident (VAST Challenge 2014). In *VAST Challenge 2014 - Honorable Mention for Effective Analytic Presentation*, 2014.

[3] Ming C. Hao, Halldór Janetzko, Sebastian Mittelstädt, Walter Hill, Umeshwar Dayal, Daniel A. Keim, Manish Marwah, and Ratnesh K. Sharma. A Visual Analytics Approach for Peak-Preserving Prediction of Large Seasonal Time Series. *Computer Graphics Forum*, 30(3):691–700, 2011.

[4] Ming C. Hao, Manish Marwah, Sebastian Mittelstädt, Halldór Janetzko, Michael Hund, Daniel A. Keim, Umeshwar Dayal, Collin Bash, Carlos Felix, Chandrakant Patel, and Meichun Hsu. Visual analytics of cyber physical data streams using spatio-temporal radial pixel visualization. *In Proceedings of SPIE 8654, Visualization and Data Analysis*, pages 865404–865412, February 2013.

[5] Ming C. Hao, Manish Marwah, Sebastian Mittelstädt, Halldór Janetzko, Daniel A. Keim, Umeshwar Dayal, Collin Bash, Carlos Felix, Chandrakant Patel, and Meichun Hsu. Exploring Cyber Physical Data Streams

Using Radial Pixel Visualizations. *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, 2012.

[6] Halldór Janetzko, Ming C. Hao, Sebastian Mittelstädt, Umeshwar Dayal, and Daniel A. Keim. Enhancing Scatter Plots Using Ellipsoid Pixel Placement and Shading. In Jr. Ralph H. Spragü, editor, *In Proceedings of the 46th Annual Hawaii International Conference on System Sciences*, pages 1522–1531. IEEE Computer Society, January 2013.

[7] Halldór Janetzko, Florian Stoffel, Sebastian Mittelstädt, and Daniel A. Keim. Anomaly Detection for Visual Analytics of Power Consumption Data. *Computer & Graphics*, 38:27–37, 2014. 23, 157, 158, 181, 210

[8] Sebastian Mittelstädt, Jürgen Bernard, Tobias Schreck, Martin Steiger, Jörn Kohlhammer, and Daniel A. Keim. Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis. *In Proceedings of the Eurographics Conference on Visualization*, pages 91–95, 2014. 19, 32, 77, 79, 87, 96

[9] Sebastian Mittelstädt, Ming C. Hao, Umeshwar Dayal, Meichun Hsu, Joseph Terdiman, and Daniel A. Keim. Advanced Visual Analytics Interfaces for Adverse Drug Event Detection. *In Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 237–244, 2014. 22, 32, 68, 69

[10] Sebastian Mittelstädt, Dominik Jäckle, Florian Stoffel, and Daniel A. Keim. ColorCAT: Guided Design of Colormaps for Combined Analysis Tasks. In *Proceedings of the Eurographics Conference on Visualization (EuroVis 2015)*, pages 115–119. The Eurographics Association, 2015. 19, 32, 58, 59, 62, 63, 65

[11] Sebastian Mittelstädt and Daniel A. Keim. Efficient Contrast Effect Compensation with Personalized Perception Models. *Computer Graphics Forum*, 34(3):211–220, 2015. 17, 18, 106, 119, 142, 144, 145, 149, 153, 155, 157, 161, 171, 173

[12] Sebastian Mittelstädt, David Spretke, Dominik Sacha, Daniel A. Keim, Bernhard Heyder, and Joachim Kopp. Visual Analytics for Critical Infrastructures. In *Proceedings of the International ETG-Congress 2013; Symposium 1: Security in Critical Infrastructures Today*, pages 1–8. VDE, 2013. 21, 72, 73

[13] Sebastian Mittelstädt, David Spretke, Dennis Thom, Dominik Jäckle, Andreas Karsten, and Daniel A. Keim. Situational Awareness for Critical Infrastructures and Decision Support. In *Proceedings of the NATO STO IST-116 Symposium on Visual Analytics*, 2013.

[14] Sebastian Mittelstädt, Andreas Stoffel, and Daniel A. Keim. Methods for Compensating Contrast Effects in Information Visualization. *Computer Graphics Forum*, 33(3):231–240, 2014. 16, 17, 18, 106, 107, 109, 113, 127, 130, 133, 135, 137, 142

[15] Sebastian Mittelstädt, Andreas Stoffel, Tobias Schreck, and Daniel A. Keim. Analysis of Local Data Patterns by Local Adaptive Color Mapping. *Presented at the IEEE Conference on Visualization (poster paper)*, 2014. 20, 181

[16] Sebastian Mittelstädt, Xiaoyu Wang, Todd Eaglin, Dennis Thom, Daniel Keim, William Tolone, and William Ribarsky. An integrated in-situ approach to impacts from natural disasters on critical infrastructures. In *Proceedings of the 48th Hawaii International Conference on System Sciences*, pages 1118–1127. IEEE, 2015. 21, 32, 72, 73, 74, 145, 157, 158, 174, 175, 233

[17] Hansi Senaratne, Sebastian Mittelstädt, Christine Jacob, and Tobias Schreck. Uncertainty Visualization for Crisis Management in Smart Grid Environments. In *Eighth International Conference on Geographic Information Science (GIScience 2014) - Workshop on Visually Supported Reasoning with Uncertainty*, 2014.

[18] Lin Shao, Sebastian Mittelstädt, Ran Goldblatt, Itzhak Omer, Peter Bak, and Tobias Schreck. StreetExplorer: Search-based exploration of urban street networks. *Submitted to the International Conference on Information Visualization Theory and Applications*, 2016. 23, 181, 212, 213

[19] Svenja Simon, Sebastian Mittelstädt, Daniel A. Keim, and Michael Sedlmair. Bridging the Gap of Domain and Visualization Experts with a Liaison. In *Eurographics Conference on Visualization (EuroVis) - Short Papers*. The Eurographics Association, 2015.

[20] Svenja Simon, Sebastian Mittelstädt, Bum Chul Kwon, Andreas Stoffel, Richard Landstorfer, Klaus Neuhaus, Anna Mühlig, Siegfried Scherer, and Daniel A. Keim. VisExpress - Visual exploration of differential gene expression data. *Information Visualization Journal (in print)*, 0:1–1, 2015.

[21] Martin Steiger, Jürgen Bernard, Sebastian Mittelstädt, Hendrik Lücke-Tieke, Daniel A. Keim, Thorsten May, and Jörn Kohlhammer. Visual Analysis of Time-Series Similarities for Anomaly Detection in Sensor Networks. *Computer Graphics Forum*, 33(3):401–410, 2014. 22, 32, 96, 97, 98

[22] Leishi Zhang, Andreas Stoffel, Michael Behrisch, Sebastian Mittelstädt, Tobias Schreck, R. Pompl, S. Weber, H. Last, and Daniel A. Keim. Visual analytics for the big data era - a comparative review of state-of-the-art commercial systems. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, 2012.

References

Adams, R. and Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6). 195

Ahn, H.-K., Cheng, S.-W., and Reinbacher, I. (2013). Maximum overlap of convex polytopes under translation. *Computational Geometry*, 46(5):552–565. 89

Anderson, B. L. and Winawer, J. (2005). Image segmentation and lightness perception. *Nature*, 434(7029):79–83. 112, 226

Anderson, M., Motta, R., Chandrasekar, S., and Stokes, M. (1996). Proposal for a standard default color space for the internet-srgb. In *Color and imaging conference*, volume 1996, pages 238–245. Society for Imaging Science and Technology. 104, 166

Andrienko, N. and Andrienko, G. (2006). *Exploratory Analysis of Spatial and Temporal Data*. Springer. 24, 27, 28, 29, 30, 35

Anscombe, F. J. (1973). Graphs in statistical analysis. *American Statistician*, 27(1):17–21. 4, 5, 231

Avery, B., Sandor, C., and Thomas, B. H. (2009). Improving spatial perception for augmented reality x-ray vision. In *IEEE Virtual Reality Conference*, pages 79–82. IEEE. 179

Bache, K. and Lichman, M. (2013). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. 79, 91, 232

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483. 88

Barten, P. G. (1999). *Contrast sensitivity of the human eye and its effects on image quality*, volume 72. SPIE press. 163, 210

Belmore, S. C. and Shevell, S. K. (2011). Very-long-term and short-term chromatic adaptation: are their influences cumulative? *Vision Research*, 51(3):362–366. 177

Bennett, M. and Quigley, A. (2011). Creating personalized digital human models of perception for visual analytics. In *User Modeling, Adaption and Personalization*, pages 25–37. Springer. 147, 148

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517. 89

Bergman, L. D., Rogowitz, B., and Treinish, L. A. (1995). A rule-based tool for assisting colormap selection. In *Proceedings of the IEEE Conference on Visualization*, pages 118–125. IEEE. 34, 35, 37, 38, 39, 43, 102, 231

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., and Wiswedel, B. (2007). KNIME: The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer. 3

Bertini, E., Girolamo, A., and Santucci, G. (2007). See what you know: Analyzing data distribution to improve density map visualization. *IEEE Symposium on Visualization (Eurographics 2007)*. 187, 203, 207, 214, 215, 216, 217

Bertini, E. and Santucci, G. (2006a). Give chance a chance: modeling density to enhance scatter plot quality through random data sampling. *Information Visualization*, 5(2):95–110. 186

Bertini, E. and Santucci, G. (2006b). Visual quality metrics. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–5. ACM. 186

Blakeslee, B. and McCourt, M. E. (1999). A multiscale spatial filtering account of the White effect, simultaneous brightness contrast and grating induction. *Vision Research*, 39(26):4361–4377. 120

Blakeslee, B., Pasieka, W., and McCourt, M. E. (2005). Oriented multiscale spatial filtering and contrast normalization: a parsimonious model of brightness induction in a continuum of stimuli including White, Howe and simultaneous brightness contrast. *Vision Research*, 45(5):607–615. 112, 226

Borland, D. and Taylor II, R. M. (2007). Rainbow Color Map (Still) Considered Harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17. 37

Brainard, D. H., Pelli, D. G., and Robson, T. (2002). Display characterization. *Encyclopedia of imaging science and technology*. 167

Brandes, U. and Pich, C. (2007). Eigensolver methods for progressive multidimensional scaling of large data. In *Proceedings of the 14th International Conference on Graph Drawing*, pages 42–53. Springer-Verlag. 85

Brehmer, M. and Munzner, T. (2013). A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385. 12, 24, 27, 28, 29, 30

Bremm, S., von Landesberger, T., Bernard, J., and Schreck, T. (2011). Assisted descriptor selection based on visual comparative data analysis. In *Computer Graphics Forum*, volume 30, pages 891–900. Wiley Online Library. 43

Brettel, H., Viénot, F., and Mollon, J. D. (1997). Computerized simulation of color appearance for dichromats. *JOSA A*, 14(10):2647–2655. 64, 90, 228

Brewer, C. A. (1994). Color use guidelines for mapping and visualization. *Visualization in modern cartography*, 2:123–148. 40

Brewer, C. A. (1996a). Guidelines for selecting colors for diverging schemes on maps. *The Cartographic Journal*, 33(2):79–86. 40, 42

Brewer, C. A. (1996b). Prediction of simultaneous contrast between map colors with Hunt's model of color appearance. *Color Research and Application*, 21(3):221–235. 108, 111

Brewer, C. A. (1997a). Evaluation of a model for predicting simultaneous contrast on color maps. *The Professional Geographer*, 49(3):280–294. 111

Brewer, C. A. (1997b). Spectral Schemes: Controversial Color Use on Maps. *Cartography and Geographic Information Systems*, 24(4):203–220. 38

Brewer, C. A. (2015). *ColorBrewer*. (visited on 22/01/2015). 34

Bruckner, S. and Möller, T. (2010). Isosurface similarity maps. In *IEEE-VGTC Symposium on Visualization 2010 (Eurographics)*, volume 29. 187

Buchanan, M. D. (1979). Effective utilization of color in multidimensional data presentations. In *23rd Annual Technical Symposium*, pages 9–18. International Society for Optics and Photonics.

Camgöz, N., Yener, C., and Güvenc, D. (2004). Effects of hue, saturation, and brightness: Part 2: Attention. *Color Research & Application*, 29(1):20–28. 52

CIE (1978). *Colorimetry: Recommendations on Uniform Color Spaces-Color Difference Equations, Psychometric Color Terms. Suppl. 2*. International Commission on Illumination, Bureau Central de la CIE. 25

Cleveland, W. S. and McGill, R. (1983). A color-caused optical illusion on a statistical graph. *The American Statistician*, 37(2):101–105. 108, 111

Crassini, B., Brown, B., and Bowman, K. (1988). Age-related changes in contrast sensitivity in central and peripheral retina. *Perception*, 17(3):315–332. 163

Cui, G., Luo, M. R., Rigg, B., Roesler, G., and Witt, K. (2002). Uniform colour spaces based on the DIN99 colour-difference formula. *Color Res. Appl.*, 27(4):282–290. 114

Curcio, C. A., Sloan, K. R., Packer, O., Hendrickson, A. E., and Kalina, R. E. (1987). Distribution of cones in human and monkey retina: individual variability and radial asymmetry. *Science*, 236(4801):579–582. 159, 173

Demiralp, C. D., Bernstein, M. S., and Heer, J. (2014). Learning perceptual kernels for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1933–1942. 227

DIN (2000). *Farbmetrische Bestimmung von Farbabständen bei Körperfarben nach der DIN 99-Formel*. Berlin: Deutsches Institut für Normung e.V. 26

Duda, R. O., Hart, P. E., et al. (1973). Pattern classification and scene analysis. *John Wiley and Sons*. 195, 196, 234

Eagleman, D. M., Jacobson, J. E., and Sejnowski, T. J. (2004). Perceived luminance depends on temporal context. *Nature*, 428(6985):854–856. 177

Ekroll, V. and Faul, F. (2009). A simple model describes large individual differences in simultaneous colour contrast. *Vision research*, 49(18):2261–2272. 146, 178

Elmqvist, N., Dragicevic, P., and Fekete, J. (2011). Color lens: Adaptive color scale optimization for visual exploration. *IEEE Transactions on Visualization and Computer Graphics*, 17(6). 188

Enroth-Cugell, C. and Shapley, R. (1973). Adaptation and dynamics of cat retinal ganglion cells. *The Journal of Physiology*, 233(2). 199

Eriksen, C. W. and Hake, H. W. (1955). Multidimensional stimulus differences and accuracy of discrimination. *Journal of Experimental Psychology*, 50(3):153. 49, 50, 54

Eskew, R. T. (2009). Higher order color mechanisms: A critical review. *Vision Research*, 49(22):2686–2704. 178

Fairchild, M. D. (2013). *Color Appearance Models*. John Wiley & Sons. 24, 26, 46, 47, 48, 111, 173

Fairchild, M. D. and Johnson, G. M. (2004). iCAM framework for image appearance, differences, and quality. *Journal of Electronic Imaging*, 13(1):126–138. 112, 118, 119, 120, 138, 140, 146, 178, 225, 233

Fairchild, M. D. and Johnson, G. M. (2005). *Meet iCAM*. (visited on 12/04/2013). 120, 121

Fekete, J.-D., Van Wijk, J. J., Stasko, J. T., and North, C. (2008). The value of information visualization. In *Information visualization*, pages 1–18. Springer. 4

Flatla, D. and Gutwin, C. (2012). SSMRecolor: improving recoloring tools with situation-specific models of color differentiation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2297–2306. ACM. 147

Flatla, D. R. and Gutwin, C. (2011). Improving calibration time and accuracy for situation-specific models of color differentiation. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 195–202. ACM. 104, 147, 178

Flatla, D. R., Reinecke, K., Gutwin, C., and Gajos, K. Z. (2013). SPRWeb: Preserving Subjective Responses to Website Colour Schemes through Automatic Recolouring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2069–2078. ACM. 147

Garner, W. R. and Hake, H. W. (1951). The amount of information in absolute judgments. *Psychological Review*, 58(6):446. 49

Gilchrist, A. (2006). *Seeing Black and White*. Oxford University Press. 112

Gilchrist, A., Kossyfidis, C., Bonato, F., Agostini, T., Cataliotti, J., Li, X., Spehar, B., Annan, V., and Economou, E. (1999). An anchoring theory of lightness perception. *Psychological Review*, 106(4):795–809. 112, 226

Giusti, A., Taddei, P., Corani, G., Gambardella, L., Magli, C., and Gianaroli, L. (2011). Artificial defocus for displaying markers in microscopy z-stacks. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1757–1764. 179

Gonzalez, R. C. and Woods, R. E. (2007). Digital image processing 3rd edition. 196

Granzier, J. J., Toscani, M., and Gegenfurtner, K. R. (2012). Role of eye movements in chromatic induction. *JOSA A*, 29(2):A353–A365. 179

Green, T. M. and Fisher, B. (2010). Towards the personal equation of interaction: The impact of personality factors on visual analytics interface interaction. In *IEEE Symposium on Visual Analytics Science and Technology (VAST 2010)*, pages 203–210. IEEE. 148

Guo, D., Chen, J., MacEachren, A. M., and Liao, K. (2006). A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1474. 42

Guo, D., Gahegan, M., MacEachren, A. M., and Zhou, B. (2005). Multivariate analysis and geovisualization with an integrated geographic knowledge discovery approach. *Cartography and Geographic Information Science*, 32(2):113–132. 42, 52

Hansen, T. and Gegenfurtner, K. R. (2013). Higher order color mechanisms: Evidence from noise-masking experiments in cone contrast space. *Journal of vision*, 13(1):26. 178

Harnad, S. (2003). Categorical perception. *Encyclopedia of cognitive science*, 67(4). 46, 47, 54

Harrower, M. and Brewer, C. (2003). Colorbrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal*, 40(1):27–37. 35, 40, 42, 43, 52, 100, 102

Healey, C. (1996). Choosing effective colours for data visualization. In *Proceedings of Visualization.*, pages 263–270. IEEE. 41

Herman, I., Marshall, M., and MelanÃ§on, G. (2000). Density functions for visual attributes and effective partitioning in graph visualization. In *IEEE Symposium on Information Visualization*. 187

Himberg, J. (2000). A SOM based cluster visualization and its application for false coloring. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 3, pages 587–592. IEEE. 43, 77

Hunt, R. W. G. and Pointer, M. R. (2011). *Measuring Colour*. John Wiley & Sons. 111

Hwang, A. D. and Peli, E. (2014). An augmented-reality edge enhancement application for google glass. *Optometry & Vision Science*, 91(8):1021–1030. 179

Isenberg, P., Dragicevic, P., Willett, W., Bezerianos, A., and Fekete, J.-D. (2013). Hybrid-Image Visualization for Large Viewing Environments. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2346–2355. 163, 164

ISMP QuarterWatch (2011). Signals for varenicline, levofloxacin and fentanyl. http://www.ismp.org/quarterwatch/pdfs/2010Q2.pdf (visited on 10/06/2013). 69

ISMP QuarterWatch (2012a). Signals for dabigatran and metoclopramide. http://www.ismp.org/quarterwatch/pdfs/2011Q1.pdf (visited on 10/06/2013). 69

ISMP QuarterWatch (2012b). Why reports of serious adverse drug events continue to grow. http://www.ismp.org/quarterwatch/pdfs/2012Q1.pdf (visited on 10/06/2013). 69

Itoh, Y. and Klinker, G. (2015). Vision enhancement: defocus correction via optical see-through head-mounted displays. In *Proceedings of the 6th Augmented Human International Conference*, pages 1–8. ACM. 148, 179

Kaski, S., Venna, J., and Kohonen, T. (2000). Coloring that reveals cluster structures in multivariate data. *Australian Journal of Intelligent Information Processing Systems*, 6(2):82–88. 19, 77, 78, 79, 82, 83, 84, 85, 232

Keim, D. (2000). Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78. 25, 38, 52, 61, 65, 95, 111, 183, 209, 211

Keim, D., Ankerst, M., and Kriegel, H. (1995). Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings of the 6th conference on Visualization*. 196, 209

Keim, D., Hao, M., Dayal, U., and Hsu, M. (2002). Pixel Bar Charts: A Visualization Technique for Very Large Multi-Attribute Data Sets. *Information Visualization (Infovis 2002)*, 1(1). 183

Keim, D. and Oelke, D. (2007). Literature fingerprinting: A new method for visual literary analysis. In *IEEE Symposium on Visual Analytics Science and Technology*. 183

Keim, D. A., Hao, M. C., Dayal, U., Janetzko, H., and Bak, P. (2010). Generalized scatter plots. *Information Visualization*, 9(4):301–311. 70, 71

Kennedy, J., Eberhart, R., and others (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pages 1942–1948. 85, 86

Kindlmann, G., Reinhard, E., and Creem, S. (2002). Face-based luminance matching for perceptual colormap generation. In *Proceedings of the Conference on Visualization*, pages 299–306. IEEE Computer Society. 38, 41, 111, 147

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., and others (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680. 125

Kopf, J. and Lischinski, D. (2011). Depixelizing Pixel Art. *ACM Trans. on Graphics*, 30(4):99:1 – 99:8. 152

Kuhbandner, C. and Pekrun, R. (2013). Joint effects of emotion and color on memory. *Emotion*, 13(3):375. 8

Kuhn, G. R., Oliveira, M. M., and Fernandes, L. A. (2008). An efficient naturalness-preserving image-recoloring method for dichromats. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1747–1754. 44

Lee, S., Sips, M., and Seidel, H.-P. (2012). Perceptually-Driven Visibility Optimization for Categorical Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, pages 1746–1757. 16, 42, 65, 125

Levkowitz, H. and Herman, G. T. (1992). The Design and Evaluation of Color Scales for Image Data. *IEEE Computer Graphics and Applications*, 12(1):72–80. 38, 41, 61, 111

Light, A. and Bartlein, P. J. (2004). The end of the rainbow? Color schemes for improved data graphics. *Transactions American Geophysical Union*, 85(40):385–391. 37

Lin, S., Fortuna, J., Kulkarni, C., Stone, M., and Heer, J. (2013). Selecting Semantically-Resonant Colors for Data Visualization. In *Computer Graphics Forum*, volume 32, pages 401–410. Wiley Online Library. 8, 11, 42, 229

Liu, Z. and Heer, J. (2014). The Effects of Interactive Latency on Exploratory Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131. 143, 145, 176

Lundström, C., Ljung, P., and Ynnerman, A. (2006). Local histograms for design of transfer functions in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(6). 187

Luo, M. R., Cui, G., and Li, C. (2006). Uniform colour spaces based on CIECAM02 colour appearance model. *Color Research & Application*, 31(4):320–330. 26, 55, 65

MacAdam, D. L. (1942). Visual sensitivities to color differences in daylight. *JOSA*, 32(5):247–273. 55, 150

Machado, G. M. and Oliveira, M. M. (2010). Real-Time Temporal-Coherent Color Contrast Enhancement for Dichromats. In *Computer Graphics Forum*, volume 29, pages 933–942. Wiley Online Library. 44

Machado, G. M., Oliveira, M. M., and Fernandes, L. A. (2009). A physiologically-based model for simulation of color vision deficiency. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1291–1298. 90

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *Transactions On Graphics*, 5(2):110–141. 12

Mahy, M., Eycken, L., and Oosterlinck, A. (1994). Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV. *Color Research & Application*, 19(2):105–121. 54, 89, 150, 197

Mannos, J. L. and Sakrison, D. J. (1974). The effects of a visual fidelity criterion of the encoding of images. *Information Theory, IEEE Transactions on*, 20(4):525–536. 163, 164, 234

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences*, 207(1167):187–217. 120

Mei, M. and Leat, S. J. (2007). Suprathreshold contrast matching in maculopathy. *Investigative Ophthalmology and Visual Science*, 48(7):3419. 163

Meylan, L., Alleysson, D., and Süsstrunk, S. (2007). Model of retinal local adaptation for the tone mapping of color filter array images. *J. Opt. Soc. Am. A*, 24(9). 199, 200, 201, 202

Moreland, K. (2009). Diverging color maps for scientific visualization. In *Advances in Visual Computing*, pages 92–103. Springer. 41

Moroney, N., Fairchild, M. D., Hunt, R. W., Li, C., Luo, M. R., and Newman, T. (2002). The CIECAM02 color appearance model. In *Proceedings of the Color and Imaging Conference*, volume 2002, pages 23–27. Society for Imaging Science and Technology. 26, 111, 118, 119, 120

Mullen, K. T. (1985). The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. *The Journal of Physiology*, 359:381. 101

Munzner, T. (2014). *Visualization Analysis and Design*. CRC Press. 5, 7, 11

Naka, K.-I. and Rushton, W. A. (1966). S-potentials from luminosity units in the retina of fish (Cyprinidae). *The Journal of physiology*, 185(3). 199

Nam, J., Ro, Y. M., Huh, Y., and Kim, M. (2005). Visual content adaptation according to user perception characteristics. *IEEE Transactions on Multimedia*, 7(3):435–445. 147

Nissen, S. E. and Wolski, K. (2007). Effect of rosiglitazone on the risk of myocardial infarction and death from cardiovascular causes. *New England Journal of Medicine*, 356(24):2457–2471. 69

Norman, D. (2002). Emotion & design: attractive things work better. *Interactions*, 9(4):36–42. 103

Oelke, D., Janetzko, H., Simon, S., Neuhaus, K., and Keim, D. (2011). Visual Boosting in Pixel-based Visualizations. *IEEE Symposium on Visualization 2011 (EuroVis)*, 30(3). 188, 203, 211

Oliveira, M. M. (2013). Towards More Accessible Visualizations for Color-Vision-Deficient Individuals. *Computing in Science & Engineering*, 15(5):80–87. 44

Palmer, S. E. (1999). *Vision Science: Photons to Phenomenology*. John Wiley & Sons. 46, 47, 48

Parraga, C. A., Roca-Vila, J., Karatzas, D., and Wuerger, S. M. (2014). Limitations of visual gamma corrections in LCD displays. *Displays*, 35(5):227–239. 178

Pelli, D. G. and Bex, P. (2013). Measuring contrast sensitivity. *Vision research*, 90:10–14. 163

Perlin, K. and Fox, D. (1993). Pad: an alternative approach to the computer interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 57–64. ACM. 188

Pizer, S., Amburn, E., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., ter Haar Romeny, B., Zimmerman, J., and Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3). 187

Pizer, S. M. and Zimmerman, J. B. (1983). Color display in ultrasonography. *Ultrasound in medicine & biology*, 9(4):331–345. 38, 41

Press, H., Teukolsky, S. A., and Vetterling, W. T. (2007). Numerical recipes: the art of scientific computing. 150

Quinlan, J. R. (1993). Combining instance-based and model-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 236–243. University of Massachusetts. 2, 3

Rheingans, P. (2000). Task-based color scale design. In *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*, pages 35–43. International Society for Optics and Photonics. 34

Roberson, D. and Davidoff, J. (2000). The categorical perception of colors and facial expressions: The effect of verbal interference. *Memory & Cognition*, 28(6):977–986. 8, 47, 229

Robertson, P. K. and O'Callaghan, J. F. (1986). The generation of color sequences for univariate and bivariate mapping. *IEEE Computer Graphics and Applications*, 6(2):24–32. 38, 41

Robinson, A. E., Hammon, P. S., and de Sa, V. R. (2007). Explaining brightness illusions using spatial filtering and local response normalization. *Vision research*, 47(12):1631–1644. 226

Rogowitz, B. and Kalvin, A. D. (2001). The Which Blair Project: a quick visual method for evaluating perceptual color maps. In *Proceedings of the IEEE Conference on Visualization.*, pages 183–556. IEEE. 41, 147

Rogowitz, B., Treinish, L., and Bryson, S. (1996). How not to lie with visualization. *Computers in Physics*, 10(3):268–273. 12, 35, 37, 38, 39, 54, 56, 102

Rogowitz, B. E. and Treinish, L. A. (1993). An architecture for rule-based visualization. In *Proceedings of the 4th Conference on Visualization'93*, pages 236–243. IEEE Computer Society. 38, 102

Ruiz, M., Bardera, A., Boada, I., Viola, I., Feixas, M., and Sbert, M. (2011). Automatic transfer functions based on informational divergence. *IEEE Transactions on Visualization and Computer Graphics*, 17(12). 187

Sajadi, B., Majumder, A., Oliveira, M. M., Schneider, R. G., and Raskar, R. (2013). Using patterns to encode color information for dichromats. *IEEE Transactions on Visualization and Computer Graphics*, 19(1):118–129. 44

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409. 56, 82

Samsel, F., Petersen, M., Geld, T., Abram, G., Wendelberger, J., and Ahrens, J. (2015). 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*, pages 703–710. ACM. 103

Schade, S. (1956). Optical and photoelectric analog of the eye. *JoSA*, 46(9):721–738. 163

Sedlmair, M., Heinzl, C., Bruckner, S., Piringer, H., and Möller, T. (2014). Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170. 16, 152

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press. 49

Shi, V., Cui, J., Troncoso, X. G., Macknik, S. L., and Martinez-Conde, S. (2013). Effect of stimulus width on simultaneous contrast. *PeerJ*, 1:146. 120, 139

Silva, S., Sousa Santos, B., and Madeira, J. (2011). Using color in visualization: A survey. *Computers & Graphics*, 35(2):320–333. 41

Steichen, B., Carenini, G., and Conati, C. (2013). User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 317–328. ACM. 148

Steichen, B., Wu, M. M., Toker, D., Conati, C., and Carenini, G. (2014). Te, Te, Hi, Hi: Eye gaze sequence analysis for informing user-adaptive information visualizations. In *User Modeling, Adaptation, and Personalization*, pages 183–194. Springer. 148

Stewart, N., Brown, G. D., and Chater, N. (2005). Absolute Identification by Relative Judgment. *Psychological Review*, 112(4):881–911. 49

Stockman, A. (2009). *Color vision mechanisms*. PhD thesis, University of Pennsylvania. 178

Stone, J. E., Gohara, D., and Shi, G. (2010). OpenCL: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering*, 12(3):66. 151

To, L., Woods, R. L., Goldstein, R. B., and Peli, E. (2013). Psychophysical contrast calibration. *Vision research*, 90:15–24. 167, 168, 169

Toker, D., Conati, C., Steichen, B., and Carenini, G. (2013). Individual user characteristics and information visualization: connecting the dots through eye tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 295–304. ACM. 148

Tominski, C., Fuchs, G., and Schumann, H. (2008). Task-driven color coding. In *Proceedings of the 12th International Conference on Information Visualisation*, pages 373–380. IEEE. 12, 28, 35, 38, 39, 43, 101, 231

Treisman, A. (1985). Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31(2):156–177. 183

Trumbo, B. E. (1981). A theory for coloring bivariate statistical maps. *The American Statistician*, 35(4):220–226. 41

Tufte, E. R. (1990). Envisioning Information. *Graphics Press*. 188

Turkowski, K. (1990). Filters for common resampling tasks. In *Graphics gems*, pages 147–165. Academic Press Prof. 152, 153, 156

Vanderbei, R. J. (2012). *2012 Presidential Election Results*. (visited on 12/05/2014). 135, 157, 158, 233

Vetro, A. (2004). MPEG-21 digital item adaptation: enabling universal multimedia access. *MultiMedia, IEEE*, 11(1):84–87. 147

von Kries, J. (1905). Die Gesichtsempfindungen. *Handbuch der Physiologie des Menschen*, 3:109–282. 146, 226

Wainer, H. and Francolini, C. M. (1980). An empirical inquiry concerning human understanding of two-variable color maps. *The American Statistician*, 34(2):81–93. 43, 76

Wang, L., Giesen, J., McDonnell, K. T., Zolliker, P., and Mueller, K. (2008). Color design for illustrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1739–1754. 41, 43, 71, 103

Wang, L. and Kaufman, A. (2012). Importance Driven Automatic Color Design for Direct Volume Rendering. *IEEE Symposium on Visualization 2012*, 31(3). 188

Wanner, F., Rohrdantz, C., Mansmann, F., Stoffel, A., Oelke, D., Krstajic, M., Keim, D. A., Luo, D., Yang, J., and Atkinson, M. (2009). *Large-scale comparative sentiment analysis of news articles*. Published: Presented at the poster session at IEEE Information Visualization Conference 2009, Atlantic City, USA. 136, 137, 233

Ware, C. (1988). Color sequences for univariate maps: Theory, experiments and principles. *IEEE Computer Graphics and Applications*, 8(5):41–49. 9, 10, 35, 37, 38, 47, 49, 50, 54, 56, 108, 109, 111, 128, 131, 132, 134, 138, 191

Ware, C. (2012). *Information visualization: perception for design*. Elsevier. 6, 42, 47, 50, 54, 61, 101, 103, 211

Ware, C. and Beatty, J. C. (1988). Using color dimensions to display data dimensions. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 30(2):127–142. 42, 76, 84, 91, 93, 95, 96, 101

Watson, A. B. and Ahumada Jr, A. J. (2005). A standard model for foveal detection of spatial contrast. *Journal of Vision*, 5:717–740. 139, 163, 164

Weber, M., Alexa, M., and Müller, W. (2001). Visualizing time-series on spirals. In *Proceedings of the IEEE Symposium on Information Visualization*, page 7. IEEE Computer Society. 73, 183

Webster, M. A., Juricevic, I., and McDermott, K. C. (2010). Simulations of adaptation and color appearance in observers with varying spectral sensitivity. *Ophthalmic and Physiological Optics*, 30(5):602–610. 159, 173

Wijffelaars, M., Vliegen, R., Van Wijk, J. J., and Van Der Linden, E.-J. (2008). Generating color palettes using intuitive parameters. *Computer Graphics Forum*, 27:743–750. 34, 35, 41

Wood, J. and Dykes, J. (2008). Spatially ordered treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1348–1355. 43

Xiao, K., Fu, C., Karatzas, D., and Wuerger, S. (2011). Visual gamma correction for LCD displays. *Displays*, 32(1):17–23. 163, 167

Young, R. A. (1987). The Gaussian derivative model for spatial vision: I. Retinal mechanisms. *Spatial Vision*, 2(4):273–293. 120