

Efficient Contrast Effect Compensation with Personalized Perception Models

Sebastian Mittelstädt and Daniel A. Keim

University of Konstanz, Germany

Abstract

Color is one of the most effective visual variables and is frequently used to encode metric quantities. Contrast effects are considered harmful in data visualizations since they significantly bias our perception of colors. For instance, a gray patch appears brighter on a black background than on a white background. Accordingly, the perception of color-encoded data items depends on the surround in the rendered visualization. A method that compensates for contrast effects has been presented previously, 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. In this paper, we provide experiments that show a significant difference in the perception of users. We introduce methods to personalize contrast effect compensation and show that this outperforms the original method with a user study. We, further, overcome the major limitation of the 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.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms

1. Introduction

Color is very effective in encoding data since it can be combined with any other visual variable to enrich data representations. Color has many use cases such as to express metric quantities, categorical data, interestingness, or guide through tree-structured data [Tdj14]. It may also be used to map complex data to single colors, which is intuitively perceived if perceptual linearity is considered in the encoding [MBS*14]. However, there is a danger as for all visual mappings. If the color design is not appropriate, it may bias the analyst. Therefore, our community provides guidelines on the effective use of color. But some issues cannot be overcome in the design, because they may only appear in the real application. Studies showed that contrast effects (see Fig. 1) can bias the user in reading color encoded data by up to 17% [War88].

There exist guidelines and rules of thumbs on how to avoid contrast effects by appropriate color design. However, contrast effects cannot be avoided before the final image is rendered since color appearance (of a pixel) depends on the surround in the image (see Fig. 1). Mittelstädt *et al.* [MSK14] presented a pixel-based optimization method that is applica-

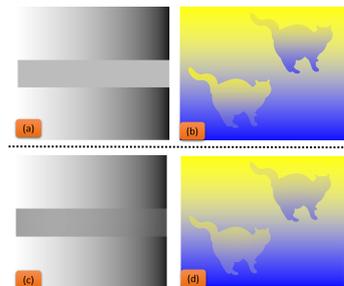


Figure 1: Examples of contrast effects. (a) The ends of the gray bar and also the cats (b) are perceived differently, but are equal. (c) and (d) show compensated results of [MSK14].

tion independent and can be used as a post-processing method to compensate for contrast effects in images or visualizations. They showed that this method can double the accuracy of users reading and comparing color encoded data.

The bias of contrast effects was also experienced in an interactive control room scenario for monitoring multiple critical infrastructures with high resolution displays [MWE*15] (see

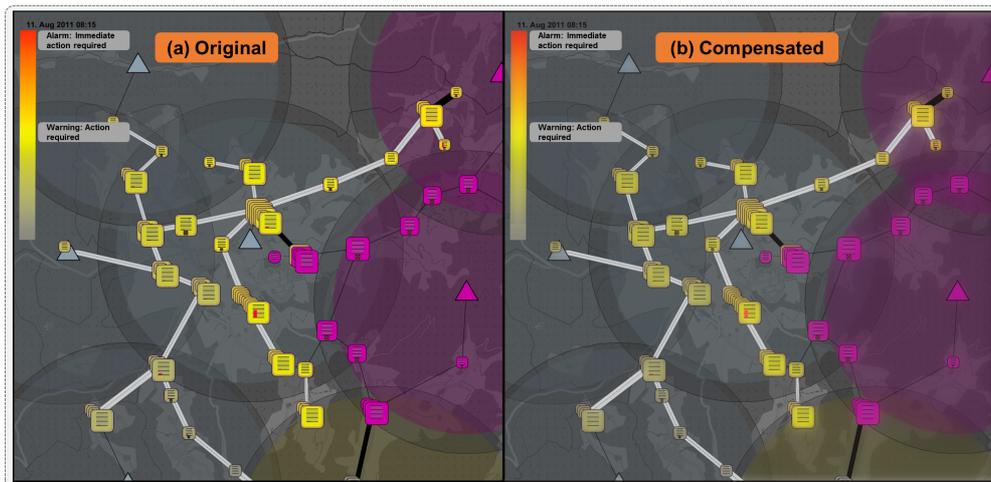


Figure 2: Overview of a smart grid [MWE*15]. Transformer stations (rectangles) are connected via power lines and linked to the communication infrastructure (triangles). While gray indicates normal operation mode, yellow elements on the screen reveal a severe situation (violet stations are destroyed due 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.

Fig. 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 to compensate for contrast effects [MSK14] takes four minutes to compute the final result in this application, which is inappropriate for interactive visualizations. We overcome this limitation by introducing an efficient algorithm that uses efficient optimization and surrogate models. Our novel method renders the high resolution images of the sample application in 360ms, which is below the latency limit of 500ms for interactions [LH14].

Another limitation is that the perception model of the method assumes optimal conditions such as D65 ambient light, a color calibrated monitor and an “average” user. We hypothesize that the perception of contrast effects is different from individual to individual. Therefore, we provide an extension to interactively personalize the method, which is possible with the efficient algorithm. We found that this significantly increases the user’s accuracy to read data values.

We claim the following 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.

2. Related Work

2.1. Colormapping Guidelines & Color Perception

Encoding data with color is a well discussed topic in the literature. Some approaches provide data-dependent [HB03] and task-dependent guidelines [BRT95] to decide for an appropriate colormap. The impact of contrast effects in information visualization was studied by Cleveland [CM83], Ware [War88], and Brewer [Bre96]. There exists methods to

reduce harmful contrast effects such as *spiral colormaps* for continuous data that vary over hues with linearly increasing intensity [War88, LH92, MJSK15]. Further, Brewer [Bre96] introduced a model to predict contrast effects and provides guidelines to avoid contrast effects for categorical colormapping. These approaches, however, cannot compensate for contrast effects because color appearance depends on the surround in the final rendering. Therefore, Mittelstädt *et al.* [MSK14] presented a post-processing method to compensate for contrast effects, which will be revisited in the following. Fairchild [Fai13] and Hunt *et al.* [HP11] offer rich discussions on the perception of color and there exist standardized color appearance models such as CIECAM02 [MFH*02] and the iCAM framework [FJ04] that can be used to computationally predict color perception. There exist, furthermore, several approaches to model different brightness effects such as the high-level (cognitive) model of Gilchrist *et al.* [GKB*99] or the low-level (physiological) model of Blakeslee *et al.* [BPM05]. There exist related models on perceptual shading presented by Schott *et al.* [SPH*09] and Soltészova *et al.* [STPV12], on depth ordering by Zheng *et al.* [ZWM13], as well as on uniform motion spaces by Birke-land *et al.* [BTV14].

2.2. Revisiting the Method for Contrast Compensation

The original method [MSK14] locally adapts each pixel and/or its background until the data encoded by the pixel is accurately perceived. This is an optimization problem since the adaption of one pixel changes the perception of surrounding pixels. Thus, the algorithm has to find a solution, which represents all pixels as accurate as possible. The necessary

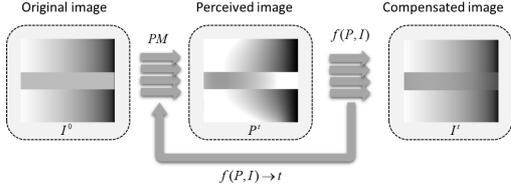


Figure 3: The algorithm computes the perception model PM , the bias with cost function f , and iteratively reduces the bias in an optimization process. Our algorithm parallelizes PM and f and also introduces a convergence threshold t .

steps of the optimization algorithm are: 1) Estimating (pixel-wise) how a human will perceive the current image based on a perception model; 2) Estimating the amount of bias by comparing the perceived and original image based on cost functions; 3) Determining the “gradient” of the bias per pixel and adapting fore- and background according to the gradient to reduce the bias. These steps are iteratively performed until the perceived bias is minimized. A schematic approach is illustrated in Fig. 3.

Perception Model. The iCAM framework [FJ04] is based on the standardized color appearance model CIECAM02. It is robust in predicting color appearance and physiological effects such as simultaneous contrast in images. According to iCAM, the color of each pixel is transformed from sRGB into the CAT02 color space. This estimates the activation of the three cone types in the human eye, which process long, medium, or short-wavelength light (LMS channels). Each LMS channel of the image is convolved with a Gaussian filter K to model the perceived surround S . Then, each pixel of the image I is locally adapted (for each channel) to the perceived surround S and to the reference white $D65$ (Eq. 1).

K depends on the spatial frequency of the image and thus, depends on the application. The original method proposes 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 spatial frequency of the image (with the highest response). The exponents ϵ_0 and ϵ_1 steer the extent of perceived contrast effects. ϵ_0 steers the effect for bright centers with dark surrounds and ϵ_1 vice versa ($\epsilon_0:0.5$; $\epsilon_1:0.6$). The constant c_1 - c_3 are used to model different ambient light conditions with chromatic adaptation for each LMS channel ($c_1: 0.94$, $c_2: 0.06$, standard: $c_3 = [94.93, 103.54, 108.72]$ [MSK14]).

$$PM(I, S, \epsilon) := (c_1 \cdot \frac{D65}{c_3 \cdot (S/I)^\epsilon} + c_2) \cdot I \quad (1)$$

$$PM'(I, S) := \begin{cases} PM(I, S, \epsilon_0), & \text{if } S > I \\ PM(I, S, \epsilon_1), & \text{else.} \end{cases} \quad (2)$$

Cost Functions. We define I as the unbiased and original input image. By our visual cognition system, we perceive the image $P = PM(I, S)$ with $P \neq I$. The difference between

P and I is the bias that misleads the analyst. The original method [MSK14] describes several cost functions that estimate the amount of perceived bias at each pixel. The most important cost function (Eq. 3) measures the color difference (Eq. 4) of the foreground pixels F between P and I in the DIN99 [CLR*02] color space (by definition, the foreground holds the data). If this cost function is zero for P , the data is accurately perceived at (and only at) each pixel. See [MSK14] for details on cost functions that remove harmful perceived gradients or preserve structures of different granularity.

$$f_1(I, P) := \frac{1}{|F|} \sum_{p \in F} \Delta E_{DIN99}(P_p, I_p) \quad (3)$$

$$\Delta E_{DIN99} := \sqrt{\Delta L_{99}^2 + \Delta a_{99}^2 + \Delta b_{99}^2} \quad (4)$$

Gradient Estimation. The approximation of the gradient for reducing contrast effects is based on two assumptions: 1) Foreground pixels represent valuable data that must be accurately represented; 2) Background pixels can be adapted to compensate for the effects on foreground pixels.

As an example, a gray foreground will appear brighter on a black background. In order to compensate for this effect, the foreground can be colored “darker”. The contrast effect of the black background will shift the “darker” foreground in the same “direction” and let it appear brighter. Thus, the gray foreground is perceived as intended. Concluding, by inverting the “direction” of the effect (e.g., darken too bright pixels) in a perceptual color space such as DIN99, the method can compensate for contrast effects at singular pixels. ΔF holds the directions of the effects, which are estimated by the difference of the original image I and the perceived image P , with $\Delta F = 0$ for background pixels (Eq. 5).

$$\Delta F := P - I \quad (5)$$

Background pixels can reduce the effects on foreground pixels by adapting in the direction of the bias. In the example, the background pixels can become brighter in order to reduce the contrast to the gray foreground. The effect on surrounding foreground pixels ΔS can be estimated by convolving the difference image ΔF with the same kernel K of the perception model (since $\Delta F = 0$ for background pixels).

$$\Delta S := \Delta F * K \quad (6)$$

The gradient to reduce contrast effects is estimated with Eq. (7), with $(\varphi_1 = \alpha, \varphi_2 = 0)$ for foreground pixels, $(\varphi_1 = 0, \varphi_2 = \alpha)$ for background pixels, and α being the step size estimated by the optimization algorithm.

$$\Delta G := -\varphi_1 \Delta F + \varphi_2 \Delta S \quad (7)$$

3. Efficient Compensation of Contrast Effects

A major limitation of the original method to compensate for contrast effects [MSK14] 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.

3.1. Why “good” solutions are “good enough”

The original method applied *line search* optimization that uses the gradient ΔG 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). The method tries to converge the cost functions to zero. However, since there is a limit for humans to perceive color differences [Mac42] (just-noticeable-difference: *JND*), 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* [PTV07], 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 Section 2.2) where $1_{\Delta E}$ approximates the *JND* for simplicity (see [MEO94] for accurate experiments).

The complexity of our algorithm can be estimated for each part. The perception model requires a convolution of the image, which has a complexity of $\mathcal{O}(2kn)$ (separable convolution), with n being the number of image pixels and k being the size of the kernel in one dimension. The estimation of the gradient does also require a second convolution with $\mathcal{O}(2kn)$. 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 complexity of the algorithm is $\mathcal{O}(c(2kn + 2kn + n)) \approx \mathcal{O}(nk^2)$.

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.

3.2. Surrogate Models

Surrogate models approximate the result of a computational model with reduced complexity or calculation on a subset of the input data [SHB*14]. 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.

3.2.1. 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

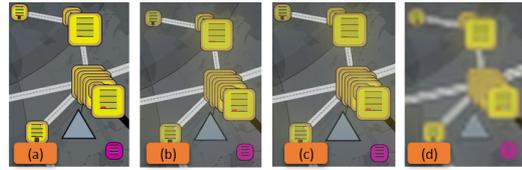


Figure 4: (a) Original image. (b) shows the compensated image without sampling in comparison to reconstruction with (c) our method and (d) the Lanczos filter method [Tur90] (both with 7px sampling interval).

result in artifacts (see Fig. 4(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 [Tur90]. Likewise in image scaling, pixels are interpolated for up or downscaling [KL11]. 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, 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 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. 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 $\mathcal{O}(nk/M^2)$.

3.2.2. Compensation with Surrogate Models

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(d)).

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.

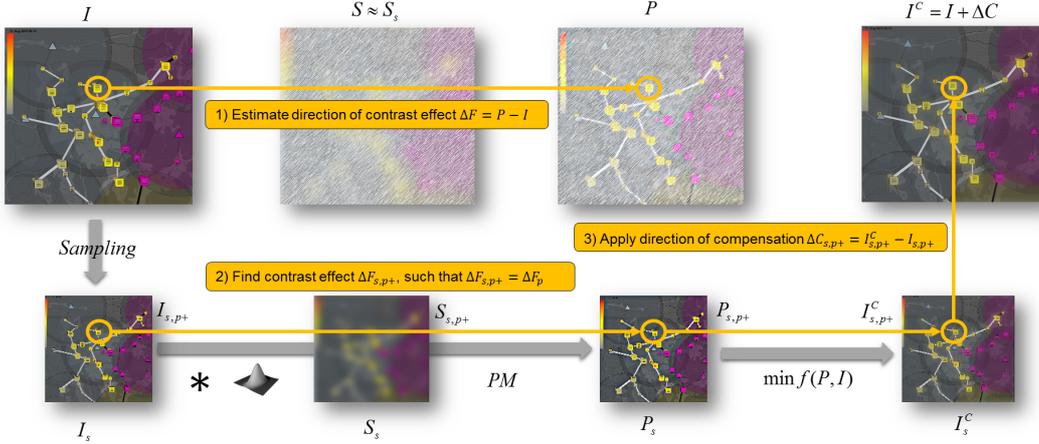


Figure 5: Contrast compensation with surrogate models. The method applies contrast compensation on the sampled image and “learns” how to compensate for contrast effects. The method determines the contrast effects for each original pixel, finds an equivalent sampled pixel in the (sampled) proximity, and applies the compensation. The algorithm adjusts the parameters automatically to provide an accurate solution without artifacts.

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. 5). 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 $\Delta F = P - I$ (including background pixels in Eq. 5).

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 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: $\cos(\Delta F_p, \Delta F_{s,p+}) > 0.99$). With A2 we can 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 $\Delta C_s = I_s^C - I_s$. The “extent” of compensation is scaled by the “extent” of the contrast effects (Eq. 8). This allows estimating the compensated image with $I^C = I + \Delta C$.

$$\Delta C_p := \frac{|\Delta F_p|}{|\Delta F_{s,p+}|} \cdot \Delta C_{s,p+} \quad (8)$$

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 these issues, 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 [Tur90]. However, in our experiments, we found that this gives poorer results than an unlimited search by our reconstruction (see Fig. 4). 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.

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 3.1). Thus, our algorithm computes the perception model on the final result (compensated image) and estimates the quality of the perceived result with the cost

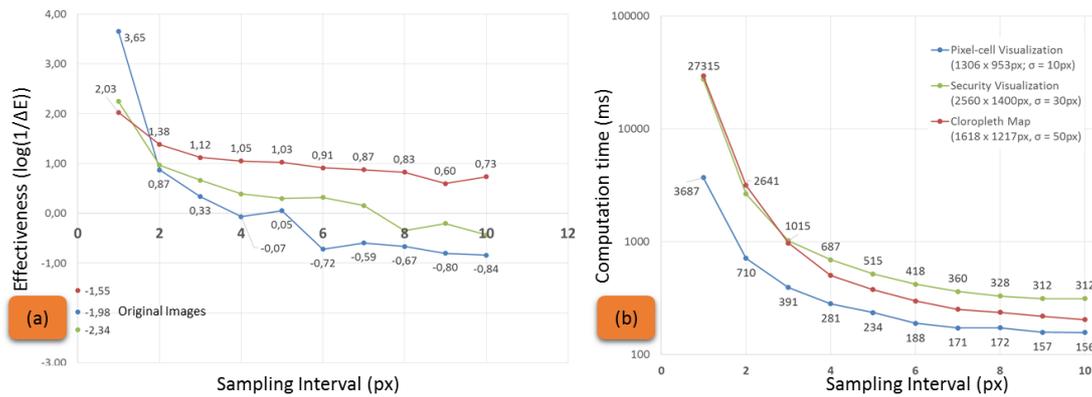


Figure 6: (a) The effectiveness of the algorithm with different sampling intervals and $t = 1_{\Delta E}$ being the quality threshold (zero line due to logarithmic scaling). At a sampling interval of 7px, the algorithm is still able to find a “good” solution for the security visualization (green). (b) The computation time decreases significantly with increasing sampling interval. With 7px sampling interval the security visualization can be rendered in 360ms.

functions of the optimization algorithm (see Section 2.1). If the quality is higher than the quality threshold, we would not perceive any difference between the optimal and the current solution (see Section 3.1). 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, with greedy algorithms or by simply decrementing the sampling interval. Section 5 discusses application depended parameterization and provides methods to support the algorithm with references to ease the search for a valid sampling interval.

3.4. Computational Evaluation

To exemplify the performance of the algorithm, we applied different sampling intervals on different types of visualizations: a security visualization [MWE*15] (Fig. 2), a choropleth map [Van12], and a pixel-based visualization [JSMK14]. All three had different resolutions and required different kernel sizes for the perception model (Fig. 6). We performed the computational evaluation on a standard laptop (Intel Core i7-4600U; Onboard Intel HD Graphics 4400).

As described in Section 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 6 reveals that the original images of all applications are suffering from the influence of contrast effects. The method without sampling (equivalent to the method of [MSK14] but GPU accelerated) is most effective. However, our 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 3.1).

4. Personalized Perception Models

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 converts the input image into the three LMS channels (for the three cone types). It predicts contrast effects by calculating Eq. 2 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 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. 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.

4.1. Methods of Interactive Personalization

Method of Adjustment. We show the user different samples of contrast effects (Fig. 7(left)). The samples consist of red, green and blue 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. 7(b)). If the user moves the left slider (ϵ_{C0}) to the right, the compensation algorithm

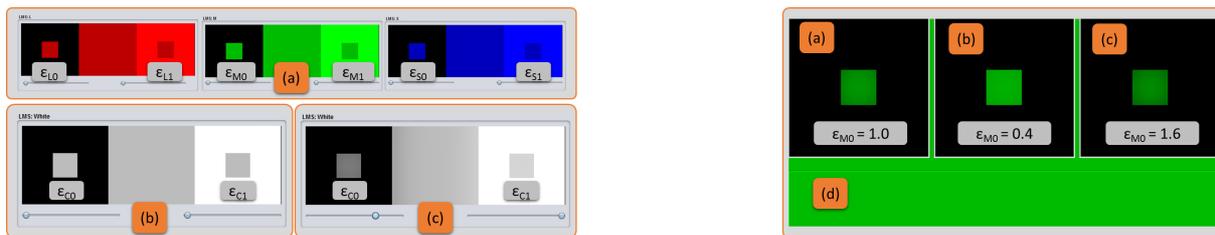


Figure 7: *Left:* (a) The user is shown samples of contrast effect for each type of cones in the human eye. The colors of the left, right, and middle patch in each sample are equal. The sliders parametrize the perception model. The user is asked to match the colors of the patches with the center and thus, personalizes the method. (b) No compensation with $\epsilon_{C0} = 0$ and $\epsilon_{C1} = 0$. (c) Compensation with $\epsilon_{C0} = 0.8$ and $\epsilon_{C1} = 1.0$. **Right:** Staircase method for personalization: The user selects the patch (a), (b), or (c) that appears most similar to the reference (d). The parameter is adapted and iteratively refined.

will (more and more) darken the gray patch (Fig. 7(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. For each exponent, we show the user an image with patches (for red, green, blue, and gray, see Fig. 7(right)) 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.

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.

4.2. Evaluation of Personalized Perception Models

Mittelstädt *et al.* [MSK14] showed that compensation of contrast effects significantly improves the accuracy of users reading and comparing color encoded data. The goal of our experiments is to evaluate if the perception of contrast effects differs between subjects, and to evaluate if personalized compensation will outperform the original method. Our hypothesis is 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 personalized perception models than with the standard perception model.*

4.2.1. 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. 8). 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.

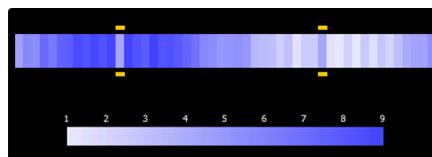


Figure 8: *The subjects had to assign the highlighted values to the value range in the color legend (here 4).*

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: 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. 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

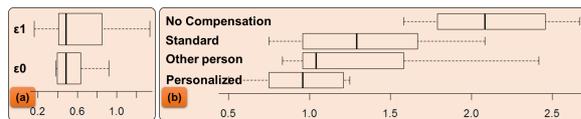


Figure 9: Experiment results. (a) Cone contrast exponents ϵ_0 and ϵ_1 of all subjects. (b) Error of subjects comparing color encoded values with different parameters for the perception model in contrast effect compensation.

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. 9 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 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 [MSK14], 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.

4.2.2. 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 [MSK14] (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 [CSP*87] and differences in spectral sensitivity [WJM10]. In our study, contrast effects caused errors in comparing color values of 23%. The error decreased to 14.2% with the standard model [MSK14] 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 [Fai13], but it remains unclear, which methods work best for contrast effect compensation.

5. 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 [MWE*15].

5.1. 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, which models the surround perception. This depends on the spatial frequency of the image and thus, depends on the application. We allow the user to load a reference image of the application and our configuration tool estimates the kernel sizes by using the approach in [MSK14] that uses Difference-of-Gaussians to determine K (see Section 2.2). 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 analysts 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 3.3). 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 files for the real 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 3.3).

5.2. 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 [MWE*15], infrastructure elements are represented by symbols, such as rectangles for transformer stations and triangles for mobile stations (see Fig. 2). 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 the community, considering issues such as perceptual linearity, intuitiveness, attention steering, and visual saliency. 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 information 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 therefore, more critical than they actually are (see Fig. 2). 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 5.1). 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 instantiate a regression model for the parameter 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. 6: $2^{2.34} JND$). 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 [LH14]. 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.

6. Discussion & Limitations

Runtime & Interaction. The runtime of the original method [MSK14] 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 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 complete. Furthermore, perception of brightness and contrast has a temporal context [EJS04, BS11]. It is not yet clear how these effects bias the analysts in the analysis process. Therefore, it remains an open question how to effectively apply contrast compensation during interactions or animations.

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 [FJ04], the mechanistic description of higher levels is still controversially discussed [Sto09]. Some studies conclude that research fails to provide evidence for higher order mechanisms in color vision at all [Esk09]. This indicates that personalization of the lowest levels might be sufficient to capture individual’s characteristics of (a)chromatic contrast perception and our experiment shows that this significantly improves the accuracy of users reading and comparing color encoded data. However, recently published results present evidence for higher order mechanisms [HG13]. We see high potential in these results for our method, but it is not yet clear how to transfer these results into a computational model for image color perception.

Future Work. The quality of compensation depends on the basis perception model, which does not model all physiological effects of color vision. Further, it is not yet clear how the gradient has to be adapted to handle imbalanced personalized exponents of cone contrasts, which currently lead to chromatic shifts in the compensated result and have to be balanced, limiting personalization. We see another challenge for collaborative environments since perception of contrast effects differs significantly between individuals.

7. Conclusion

In this paper, we present an efficient algorithm to compensate for contrast effects and provide a computational evaluation of the effectiveness in comparison to the original method [MSK14]. The performance gains make 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 of users reading and comparing color encoded data.

Acknowledgments

We would like to thank Svenja Simon and Dominik Jäckle for the great support in restructuring and revising the paper.

References

- [BPM05] BLAKESLEE B., PASIEKA W., MCCOURT M. E.: Oriented multiscale spatial filtering and contrast normalization: a parsimonious model of brightness induction in a continuum of stimuli including white, hove and simultaneous brightness contrast. *Vision Research* 45, 5 (2005), 607–615. 2
- [Bre96] BREWER C. A.: Prediction of simultaneous contrast between map colors with hunt's model of color appearance. *Color Res. Appl.* 21, 3 (1996), 221–235. 2
- [BRT95] BERGMAN L. D., ROGOWITZ B., TREINISH L. A.: A rule-based tool for assisting colormap selection. In *Proc. of the IEEE Conference on Visualization* (1995), IEEE, pp. 118–125. 2
- [BS11] BELMORE S. C., SHEVELL S. K.: Very-long-term and short-term chromatic adaptation: are their influences cumulative? *Vision Research* 51, 3 (2011), 362–366. 9
- [BTV14] BIRKELAND A., TURKAY C., VIOLA I.: Perceptually uniform motion space. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (2014), 1542–1554. 2
- [CLR*02] CUI G., LUO M. R., RIGG B., ROESLER G., WITT K.: Uniform colour spaces based on the DIN99 colour-difference formula. *Color Res. Appl.* 27, 4 (2002), 282–290. 3
- [CM83] CLEVELAND W. S., MCGILL R.: A color-caused optical illusion on a statistical graph. *The American Statistician* 37, 2 (1983), 101–105. 2
- [CSP*87] CURCIO C. A., SLOAN K. R., PACKER O., HENDRICKSON A. E., KALINA R. E.: Distribution of cones in human and monkey retina: individual variability and radial asymmetry. *Science* 236, 4801 (1987), 579–582. 8
- [EJS04] EAGLEMAN D. M., JACOBSON J. E., SEJNOWSKI T. J.: Perceived luminance depends on temporal context. *Nature* 428, 6985 (2004), 854–856. 9
- [Esk09] ESKEW R. T.: Higher order color mechanisms: A critical review. *Vision Research* 49, 22 (2009), 2686–2704. 9
- [Fai13] FAIRCHILD M. D.: *Color Appearance Models*. John Wiley & Sons, 2013. 2, 8
- [FJ04] FAIRCHILD M. D., JOHNSON G. M.: iCAM framework for image appearance, differences, and quality. *Journal of Electronic Imaging* 13, 1 (2004), 126–138. 2, 3, 9
- [GKB*99] GILCHRIST A., KOSSYFIDIS C., BONATO F., AGOSTINI T., CATALIOTTI J., LI X., SPEHAR B., ANNAN V., ECONOMOU E.: An anchoring theory of lightness perception. *Psychological Review* 106, 4 (1999), 795–809. 2
- [HB03] HARROWER M., BREWER C.: Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal* 40, 1 (2003), 27–37. 2
- [HG13] HANSEN T., GEGENFURTNER K. R.: Higher order color mechanisms: Evidence from noise-masking experiments in cone contrast space. *Journal of vision* 13, 1 (2013), 26. 9
- [HP11] HUNT R. W. G., POINTER M. R.: *Measuring Colour*. John Wiley & Sons, 2011. 2
- [JSMK14] JANETZKO H., STOFFEL F., MITTELSTÄDT S., KEIM D. A.: Anomaly Detection for Visual Analytics of Power Consumption Data. *Computers & Graphics* 38 (2014), 27–37. 6
- [KL11] KOPF J., LISCHINSKI D.: Depixelizing pixel art. *ACM Trans. on Graphics* 30, 4 (2011), 99:1 – 99:8. 4
- [LH92] LEVKOWITZ H., HERMAN G. T.: The design and evaluation of color scales for image data. *IEEE Comput. Graph. Appl.* 12, 1 (1992), 72–80. 2
- [LH14] LIU Z., HEER J.: The effects of interactive latency on exploratory visual analysis. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (2014), 2122–2131. 2, 9
- [Mac42] MACADAM D. L.: Visual sensitivities to color differences in daylight. *JOSA* 32, 5 (1942), 247–273. 4
- [MBS*14] MITTELSTÄDT S., BERNARD J., SCHRECK T., STEIGER M., KOHLHAMMER J., KEIM D. A.: Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis. In *Proc. of the Eurographics Conference on Visualization (EuroVis 2014: Short Papers)* (2014), 91–95. 1
- [MEO94] MAHY M., EYCKEN L., OOSTERLINCK A.: Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV. *Color Res. Appl.* 19, 2 (1994), 105–121. 4
- [MFH*02] MORONEY N., FAIRCHILD M. D., HUNT R. W., LI C., LUO M. R., NEWMAN T.: The CIECAM02 color appearance model. In *Proc. of the Color and Imaging Conference* (2002), vol. 2002, pp. 23–27. 2
- [MJSK15] MITTELSTÄDT S., JÄCKLE D., STOFFEL F., KEIM D. A.: ColorCAT: Guided design of colormaps for combined analysis tasks. In *Proc. of the Eurographics Conference on Visualization (EuroVis 2015: Short Papers)* (2015). 2
- [MSK14] MITTELSTÄDT S., STOFFEL A., KEIM D. A.: Methods for compensating contrast effects in information visualization. *Comput. Graph. Forum* 33, 3 (2014), 231–240. 1, 2, 3, 6, 7, 8, 9
- [MWE*15] MITTELSTÄDT S., WANG X., EAGLIN T., THOM D., KEIM D. A., TOLONE W., RIBARSKY W.: An integrated in-situ approach to impacts from natural disasters on critical infrastructures. In *Proc. of the IEEE 48th Annual Hawaii International Conference on System Sciences* (2015). 1, 2, 6, 8
- [PTV07] PRESS H., TEUKOLSKY S. A., VETTERLING W. T.: Numerical recipes: the art of scientific computing. 4
- [SHB*14] SEDLMAIR M., HEINZL C., BRUCKNER S., PIRINGER H., MÖLLER T.: Visual parameter space analysis: A conceptual framework. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (2014), 2161–2170. 4
- [SPH*09] SCHOTT M., PEGORARO V., HANSEN C., BOULANGER K., BOUATOUCH K.: A directional occlusion shading model for interactive direct volume rendering. *Comput. Graph. Forum* 28 (2009), 855–862. 2
- [Sto09] STOCKMAN A.: *Color vision mechanisms*. PhD thesis, University of Pennsylvania, 2009. 9
- [STPV12] SOLTESZOVA V., TURKAY C., PRICE M. C., VIOLA I.: A perceptual-statistics shading model. *IEEE Trans. Vis. Comput. Graphics* 18, 12 (2012), 2265–2274. 2
- [TdJ14] TENNEKES M., DE JONGE E.: Tree colors: Color schemes for tree-structured data. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (2014), 2072–2081. 1
- [Tur90] TURKOWSKI K.: Filters for common resampling tasks. In *Graphics gems* (1990), Academic Press Prof., pp. 147–165. 4, 5
- [Van12] VANDERBEI R. J.: 2012 presidential election results, 2012. (visited on 12/05/2014). URL: <http://www.princeton.edu/~rvdb/JAVA/election2012/>. 6
- [War88] WARE C.: Color sequences for univariate maps: Theory, experiments and principles. *IEEE Comput. Graph. Appl.* 8, 5 (1988), 41–49. 1, 2
- [WJM10] WEBSTER M. A., JURICEVIC I., MCDERMOTT K. C.: Simulations of adaptation and color appearance in observers with varying spectral sensitivity. *Ophthalmic and Physiological Optics* 30, 5 (2010), 602–610. 8
- [ZWM13] ZHENG L., WU Y., MA K.-L.: Perceptually-based depth-ordering enhancement for direct volume rendering. *IEEE Trans. Vis. Comput. Graphics* 19, 3 (2013), 446–459. 2