ENERGY AWARE COLOUR MAPPING FOR VISUALIZATION

by

Johnson Chuang

B.Sc. Honours, University of British Columbia, 2006

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Summer 2009

APPROVAL

Name: Johnson Chuang

Degree: Master of Science

Title of Thesis: Energy Aware Colour Mapping for Visualization

Examining Committee: Dr. Hao Zhang
Chair

Dr. Torsten Möller
Senior Supervisor
Associate Professor, Computing Science

Dr. Daniel Weiskopf
Supervisor
Professor, Computer Science
Universität Stuttgart, Germany

Dr. Mark S. Drew
Examiner
Professor, Computing Science

Date Approved: June 23, 2009

Declaration of
Partial Copyright Licence

Abstract

We present a design technique for colours that lower the energy consumption of the display device. Our approach relies on a screen space variant energy model. Guided by perceptual principles, we present three variations of our approach for finding low energy, distinguishable, iso-lightness colours. The first is based on a set of discrete user-named (categorical) colours, which are ordered according to energy consumption. The second optimizes for colours in the continuous CIELAB colour space. The third is hybrid, optimizing for colours in select CIELAB colour subspaces that are associated with colour names. We quantitatively compare our colours with a traditional choice of colours, demonstrating that approximately 45 percent of the display energy is saved. The colour sets are applied to 2D visualization of nominal data and volume rendering of 3D scalar fields. A new colour blending method for volume rendering which preserves hues further improves colour distinguishability.

Keywords: colour mapping; colour perception; display energy; volume rendering; colour blending; image compositing; perceptual transparency; illustrative visualization

Subject terms: image processing; visualization data processing; computer graphics

Acknowledgments

I would like to thank both my supervisors Torsten Möller and Daniel Weiskopf for guiding me through these two and a half years of research and studies. Their knowledge and expertise in their fields have helped me gain a greater understanding and appreciation for research work.

I also want to thank all my labmates from the Gruvi Lab. They are a fun and brainy bunch. As well, I am grateful for all the sources of funding – teaching assistantships, research assistantships, university awards and fellowships, and NSERC.

Finally, special thanks to my family and friends for their continual love and support over the years.

Part of this thesis appeared as [7]. ©Eurographics Association 2009. Reproduced with kind permission of the Eurographics Association.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

An increasing use of electrical energy in today's technologically-driven world is leading to a growth in greenhouse gas emissions [29]. As the world becomes increasingly aware of its environment, there is also a trend for IT industries to look into greener computing [44]. Green computing encourages the practice of using computing resources efficiently in such a way as to create minimal impact on the environment. A key green objective, therefore, is to reduce the energy consumption of using computers and their related subsystems.

The mobile device industry, one of the fastest growing in consumer electronics, is also seeing a trend to go green. The display is one of the major power consumers in modern computer systems, using up to 38 percent of the total system power in PCs and up to 50 percent in mobile devices [28, 37]. Thin film transistor liquid crystal displays (TFT LCDs) currently dominate the display market. One of the main components of a TFT LCD screen is the display backlight, which is often regarded as the primary consumer of display energy. Previous studies show that TFT LCDs are not very energy efficient as the electrical-to-light energy conversion has losses exceeding 80 percent [17]. Energy efficiency especially is a priority for mobile devices because they are typically powered by batteries and have limited energy resources. Furthermore, there is a trend for battery capacities to be increasing at a much slower rate than the energy requirements for mobile applications [26]. Emerging display technologies such as organic light-emitting diode (OLED) displays promise to be more energy efficient than their TFT LCD counterpart [12]. With the increased sophistication and rising energy demands of multimedia applications, it is advantageous to

explore energy saving techniques for these upcoming displays.

Compute-intensive applications like volume visualization, previously feasible only on desktop machines, are also beginning to make their way to mobile devices [27]. One of the fundamental ways to visualizing and communicating complex information is by labeling data using colours. For example, in volume visualization, users often use colours for tasks like grouping and labeling of data. Furthermore, labeling different regions on a map using colour is common in order to study and communicate weather data, census data, or a number of other geospatial information. Previous work does not consider the energy cost of colour for displays. Colour, however, plays an important role in energy consumption for emerging display technologies.

Our goal is to spark awareness of energy consumption of colours for emerging displays. Specifically, we study the effect of colour on the display energy in scientific applications like volume visualization, to more general applications like colour labeling of maps. We present three design techniques for colours that lower energy consumption. All three approaches apply an optimization process to minimize energy, and they take into account different perceptual aspects such as perceptual colour difference or categorization and naming of colours. We also present a hue-preserving colour blending method that can be used in volume rendering to improve colour distinguishability and further lower display energy consumption. Our work could benefit emerging display technologies, for example, by extending the battery life of mobile devices and lowering the energy cost of desktop machines.

1.2 Background

1.2.1 Power Management and Energy Awareness in Current and Future Display Technologies

In modern desktop computers and mobile devices, the display subsystem is one of the major consumers of electrical energy. For example, TFT LCDs, which dominate the current display market, typically consume 30 to 50 percent of battery power on mobile phones [37]. Most of that power is used by the cold cathode fluorescent lamp (CCFL) which provides the necessary background lighting to illuminate the LCD. In the energy pathway of the LCD display, no more than 20 percent of the electrical energy from the battery is converted to light form in the CCFL. Furthermore, as light from the CCFL passes through diffusers, polarizers

and colour filters in the LCD display, at least 50 percent of light energy is absorbed and lost [17, 40]. Most previous work on TFT LCDs, therefore, reduce energy consumption by decreasing the CCFL backlight intensity, and compensate for the reduced LCD luminance by increasing its transmittance [38, 5, 6]. Other ways to reduce energy consumption include varying the refresh rate of the LCD display or adopting a new pixel organization based on colour depth [6].

The key to enabling an energy efficient display design at the hardware level is to facilitate variable power output across individual subportions of the screen [15, 34]. Such a screen is *energy aware*, or *energy adaptive*, because it adapts its display light output according to the brightness intensity of the displayed image. Current main stream TFT LCD technology is not energy-adaptive, because it relies on a single constant backlight to illuminate the whole screen. The technological role model for our approach is a line of emerging displays that use space-variant lighting—e.g., Sony's OLED displays, Samsung's 9-series "local dimming" TVs, and even high dynamic range (HDR) displays that use space-variant background lighting in combination with a high-resolution TFT LCD panel [36]. This kind of HDR display is expected to hit the market in the near future, for example, in the form of Dolby HDR Video technology, which originated from BrightSide technologies [8].

1.2.2 Colour Maps and Human Visual Perception

Mapping scalar values to colour (facilitated via colour maps) is one of the most common approaches to data visualization. Two types of colour maps can be distinguished: discrete and continuous. A discrete colour map is a one-to-one mapping of discrete data values to colour, and is often used for the visualization of nominal (categorical) data. Continuous colour maps, on the other hand, define a continuous colour range. In our work, we assume that an effective colour palette is provided for the visualization of nominal data, i.e., for clearly separable elements or regions in the visualization.

Colour information arriving at the human eye is split into 3 channels: 1 achromatic and 2 chromatic [19]. The achromatic channel carries luminance information and, amongst others, is responsible for motion detection and shape perception. In contrast, the chromatic channels carry colour information useful for visual grouping and labeling. Since we focus on discrete colour maps for labeling and grouping, we target colours that are constant in lightness (i.e., iso-lightness) and only vary in colour. At this point, we make the distinction between lightness and luminance. Lightness (typically denoted L^*) is the *perceived* brightness of a

colour subject to the human perceptual system. On the other hand, luminance (denoted L) is the corresponding measurable physical quantity (in cd/m^2). Iso-lightness colours avoid unwanted effects related to non-iso-lightness colour maps for nominal data, such as emphasizing certain colour-mapped regions, which might be caused by different lightness, and introducing a natural ordering according to lightness.

The design of effective colour maps is a difficult task and often requires multiple iterations. In most applications, users have the freedom to manually specify their own colour maps or choose from an existing set of predefined ones. Although using a predefined colour map may be convenient, it may lead to faulty interpretations of the underlying data by introducing artifacts or non-existent features. For example, the popular rainbow colour map can be confusing, obscuring, and actively misleading [3]. There are tools to ease the creation of effective colour maps. For example, the ColorBrewer system uses a set of guidelines to generate a collection of colour maps for discrete data [14]. Earlier approaches to colour map design include effective colour maps based on colour distance and colour category [16], dynamic colour map creation for data exploration [35], and guidelines for univariate colours [42]. The particular problem of specifying iso-lightness colour maps, which are normally difficult to create without a calibrated monitor, can be addressed using a face-based approach [22]. There are also studies in colour design techniques for users with colour vision deficiencies [23].

All of the previous work on colour map design and colour perception neglect energy requirements of the display. One reason for this is that older displays (CRT, LCD, etc.) use approximately constant energy for displaying any colour. On the contrary, our work focuses on emerging display technologies that support variable light output across the screen. Variable light output then results in variable energy consumption across the screen, depending on the intensity of the displayed image. Therefore, this paper combines existing knowledge of perceptually motivated colour map design with new low-energy requirements.

1.3 Related Work

In a related work, Zhong and Jha study the energy characteristics of various graphical user interface components for a TFT LCD handheld device [45]. They show that for a device using a single backlight, energy variations can come from different colours being displayed. They also show that user interaction with the GUI can affect energy consumption. In one

experiment, the colour of the screen is changed from black to different colours, resulting in different energy usage. Vallerio et al. also present design techniques for energy efficient graphical user interfaces [39]. One of their techniques for saving energy is to modify the GUI colour scheme.

Emerging display technology that use space variant lighting consumes less energy compared to traditional displays using a single backlight. Although they are already lower in energy consumption, further energy savings can be achieved by the use of dark colours. There are some work on energy efficient graphical user interfaces for emerging display technology. Iyer et al. and Ranganathan et al. propose the use of *dark windows* to save energy in inactive screen areas on OLED displays [18, 34]. They achieve lower energy cost by half-dimming, full-dimming, gray scaling, or green scaling parts of the screen. However, as Harter et al. show in a user study, not all energy efficient user interfaces are acceptable by users [15]. There is a trade-off between screen content readability and the amount of energy savings.

1.4 Contributions

The main contributions of this thesis are:

- We present an energy model for emerging displays that use space-variant lighting. An energy model will allow us to quantitatively estimate and compare the energy for displaying colours and images.
- We present 3 colour design techniques that lower the energy consumption of the display device. In particular, we show that up to 45 percent of energy can be saved for applications in visualization.
- We present a new colour blending model that preserves hue. Hue-preserving colour blending can be used in volume rendering to improve colour distinguishability and further lower display energy consumption.

Please note that all images in this thesis are designed for monitor display, not for good reproduction in print. Therefore, the images are best viewed on screen.

Chapter 2

Screen Space Variant Energy Model

Displays with screen space variant control of light output, such as HDR displays [36] or OLED displays, are the basis for our model of energy estimation. Emerging display technologies make use of non-uniform backlights to vary power consumption across the screen, thereby lowering energy cost. We model the non-uniformity of such backlighting as a set of non-overlapping tiles covering the screen. The tiles are arranged in a rectangular grid, and each can be imagined as an independent backlight responsible for providing adequate lighting for the pixels within it. Although other grid layouts such as hexagonal configurations for LED-driven displays or even overlapping tiles might appear in physical devices, the qualitative results will be similar to the rectangular grid approach.

We now analyze the energy requirement within one tile. The white background lighting is the main consumer of energy. White light is filtered through the TFT LCD panel to produce different amounts of red, green, and blue colours. In general, the energy requirement is proportional to the number of "on" pixels and the brightness intensity of their R, G, and B components [18, 39]. The optimal background lighting therefore only needs to be as large as the maximum of the linear, non-gamma-corrected R, G, or B values within the tile. We use linear RGB (instead of gamma-corrected RGB, or sRGB) because they are a good approximation of the actual radiative power emitted by the display [5].

Figure 2.1 illustrates our approach of calculating the optimal background lighting. We partition the screen into a rectangular grid of $w \times h$ tiles. Within each tile i we have pixels

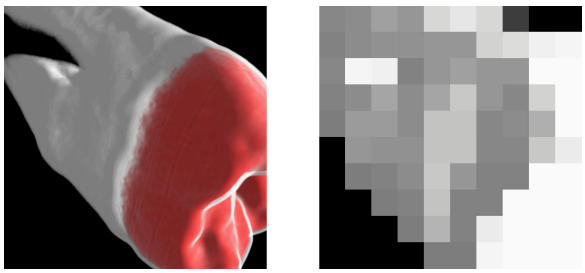


Figure 2.1: The energy required to display an image (left) is calculated by taking the maximum of the R, G, and B values within each tile. The right image illustrates a partitioning of the screen into 10×10 tiles, with shades of gray going from black to white representing the maximal R, G, B portion within that tile, estimating the energy cost.

P with red, green, and blue values in the range of 0 to 1. The maximum colour components from each tile are summed up and normalized, to obtain a relative measure of energy E_{display} between 0 and 1:

$$E_{\text{display}} = \frac{1}{wh} \sum_{i=1}^{wh} \max_{P \in \text{Tile } i} (\max(P_{\text{red}}, P_{\text{green}}, P_{\text{blue}})) \tag{2.1}$$

Equation 2.1 can be used to estimate the energy cost for displaying colour images, and subsequently to optimize for energy aware colour sets. We want to point out that taking the maximum RGB value within a tile, and then using that as energy, is only a coarse approximation for displays with multi-panel backlights (e.g., HDR displays as modelled in Figure 2.1). More complex methods for energy calculation can be used, but nevertheless Equation 2.1 is a good start and at least provides a means for qualitative comparison on real-world displays. For example, on an OLED display with separate R, G, and B primaries, a more appropriate measure of energy might be the sum of the R, G, and B components.

Chapter 3

Energy Aware Colours

Our goal is to determine a set of iso-lightness colours that are (1) easy to distinguish and (2) associated with low energy when displayed. We formulate this goal as the optimization of display energy under the constraint of good perceptual distinguishability.

3.1 Discrete Optimization

Our first optimization approach is discrete: from a set of M iso-lightness and distinguishable colours, a subset of N colours ($N \leq M$) is chosen so that the sum of the energies associated with the chosen colours is the minimum of any subset of N colours. The optimization algorithm is simple. First, the energy of each colour is computed. Second, colours are sorted according to the energy in ascending order. Third, the first N colours are picked from the sorted list.

The open question is how the set of M adequate input colours is chosen. We propose to adopt named (categorical) colours because they are sufficiently distinct and they exhibit proven perceptual benefits. For example, Kawai et al. [21] suggest that the time it takes to distinguish multiple colours depends partly on their named colour region. In a user study, Healey [16] confirms that the name of the colour region, or colour category, indeed affects the performance of colour target identification. Our discrete design technique for low energy colours makes use of this knowledge and examines colour categories and their energy performance. We further constrain our design to iso-lightness colours, which are not automatically guaranteed by named colours.

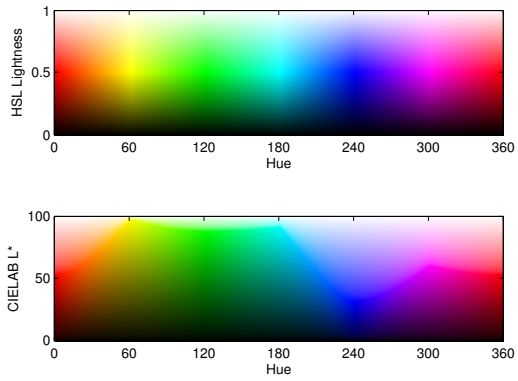


Figure 3.1: Top: Fully-saturated HSL colours. Bottom: Fully-saturated HSL colours transformed to match CIELAB's measure for lightness. Some of the colours with higher lightness values look washed-out because they are out of the monitor's gamut.

For the colour computations, we employ a perceptual colour model that has the achromatic and chromatic colour dimensions decoupled. This allows us to use the achromatic part for lightness, and the chromatic part for colour category. The CIELAB colour space [11] is a widely-used colour space designed to be perceptually uniform. It has an achromatic axis L^* for lightness and two chromatic axes a^* and b^* for red-green and yellow-blue colour components. CIELAB is the basic colour model of this paper—despite a few shortcomings compared with more advanced colour appearance models [11].

While CIELAB can aid in choosing iso-lightness colours, it lacks support for directly specifying hue, making it difficult to choose specific colour tones. LCH_{ab}, which specifies CIELAB colours using cylindrical coordinates (chroma and hue), can be used to predict colour hues in CIELAB [11]. However, because of the lack of hue uniformity, some hue slices in LCH_{ab} actually contain slightly varying colour hues. For example, a slice from the blue hue plane may contain tints of purple. Therefore, as an alternative, the HSL colour space can be applied to select hues. The only downside is that it is not perceptually uniform, so iso-lightness colour picking in HSL is not straightforward. Our solution is to transform the lightness axis in HSL to match the L^* axis in CIELAB. This gives us HSL_{LAB}, which also has 3 axes: hue, lightness, and saturation. Hue and saturation remain the same as in HSL, but lightness is now more perceptually uniform. Figure 3.1 shows a slice from a fully-saturated HSL before and after the transformation.

The number of easily distinguishable colours is small. For example, in his study of categorical colours, Healey [16] finds that 7 distinct iso-lightness colours is the maximum number of colours that can be displayed at one time without decreasing the performance of quick and accurate colour identification. He provides percentages of how likely observers will give a particular colour name to a displayed Munsell colour. Based on his findings, the colour names with high percentages of being assigned to a Munsell colour are ranked (in decreasing order): green, blue, orange, purple, red, yellow, aqua, pink, brown, and magenta. We choose the 6 most highly ranked colours (green, blue, orange, purple, red, yellow) with HSL_{LAB} hue angles 120, 240, 30, 292, 0, and 60 degrees, respectively. Saturation is set to maximum to facilitate good distinguishability.

Figure 3.2 shows the energy consumption of these 6 colours with respect to increasing lightness. The energy of a colour is the maximum of its linear R, G, or B components, as described in Chapter 2. Finally we sort the colours by increasing energy cost to obtain the energy aware categorical colour palette in Figure 3.3. Users can pick distinguishable iso-lightness colours with increasing energy cost by choosing colours from bottom to top.

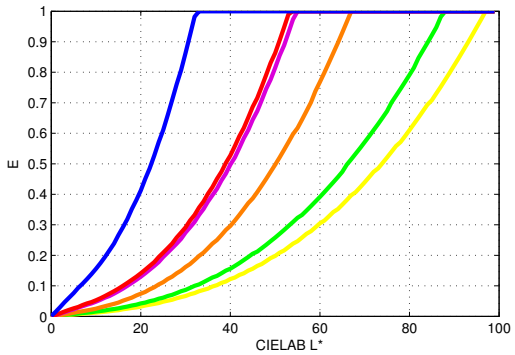


Figure 3.2: Plot of energy vs. lightness for the 6 categorical colours (left to right: blue, red, purple, orange, green, yellow). ©Eurographics Association 2009. Reproduced with kind permission of the Eurographics Association.

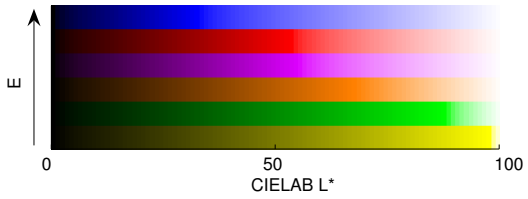
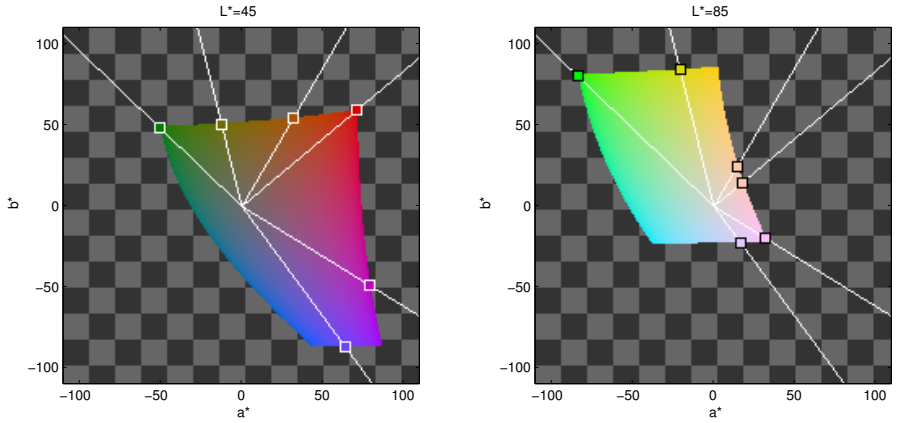


Figure 3.3: The 6 categorical colours with varying lightness sorted by increasing energy cost. ©Eurographics Association 2009. Reproduced with kind permission of the Eurographics Association.



(a) (b)

Figure 3.4: Representative LCH_{ab} hue angles for (clockwise) green, yellow, orange, red, purple, and blue, shown as white lines on CIELAB iso-lightness colour slices. (a) Medium lightness green, yellow, and orange, although low in energy, are not very distinguishable from each other because of their relative closeness. (b) High lightness orange, red, purple, and blue have very low perceptual difference and tend towards white.

While the named colours provide a reasonable choice of energy aware colours, they are not optimal when the whole available colour space is considered. First, the hues of the categorical colours are fixed and, therefore, the optimization cannot shift the colours around in colour space to achieve better perceptual distinguishability or lower energy consumption if not all M colours are needed. Second, the categorical colours are not completely evenly distributed along the boundary of the colour gamut. Figure 3.4(a) illustrates that, for example, the perceptual distance between blue and green is much larger than between green and yellow. Even colour placement would lead to better distinguishability. Third, the issue of the fixed hues is aggravated for high lightness—when colours are close to the upper lightness limit, they tend to become less saturated (shifted towards white, due to the restricted gamut of the monitor), as demonstrated in Figure 3.4(b).

3.2 Continuous Optimization

In the discrete optimization of energy aware colours, we constrained ourselves to fully-saturated colours from non-overlapping categorical colour regions. These constraints essentially restricted us to a single level of saturation and a small subset of all possible hues. We now loosen those constraints on saturation and hue, and again proceed to find sets of energy aware, distinguishable, iso-lightness colours.

A perceptual colour space like CIELAB fits our needs for distinguishability and iso-lightness. CIELAB is designed to be perceptually uniform, meaning that a perturbation to a component value produces an equal magnitude of change in visual difference. The perceived colour difference between any two colours in CIELAB can therefore be computed by treating the $L^*a^*b^*$ components as points in 3D space and taking their Euclidean distance [11]. Although CIELAB is not the most advanced colour system and does not incorporate sophisticated colour-appearance effects, it is a well-established perceptually uniform colour system. The following discussion is not restricted to CIELAB and could be immediately adopted for any other colour system that supports the computation of a measure of lightness and of perceptual difference.

We formulate our goal of finding energy aware, distinguishable, iso-lightness colours as an optimization problem with 3 input parameters: the number of colours N, the level of lightness L^*, and a minimum perceptual colour distance d (and hence, distinguishability) that must be enforced between every colour pair. Typically, d is provided by the user who wants to achieve a certain separation of colours.

The goal is to minimize the maximum display energy under the constraint that colours are at least a distance d apart from each other in CIELAB color space. Following a soft constraint implementation (i.e. penalizing colours that are closer than distance d), we propose the following cost function E_{cost} to measure the relative energy of N colours chosen from the iso-lightness colour slice. Let $C = \{C_1, C_2, ..., C_N\}$ be the N colours, then the cost is

$$E_{\text{cost}}(C) = E_{\max}(C) + k \sum_{\substack{C_i, C_j \in C \\ C_i \neq C_j}} E_{\text{penalty}}(C_i, C_j) \tag{3.1}$$

where

$$E_{\max}(C) = \max_{1 \leq i \leq N} (\max(C_{i,\text{red}}, C_{i,\text{green}}, C_{i,\text{blue}})) \tag{3.2}$$

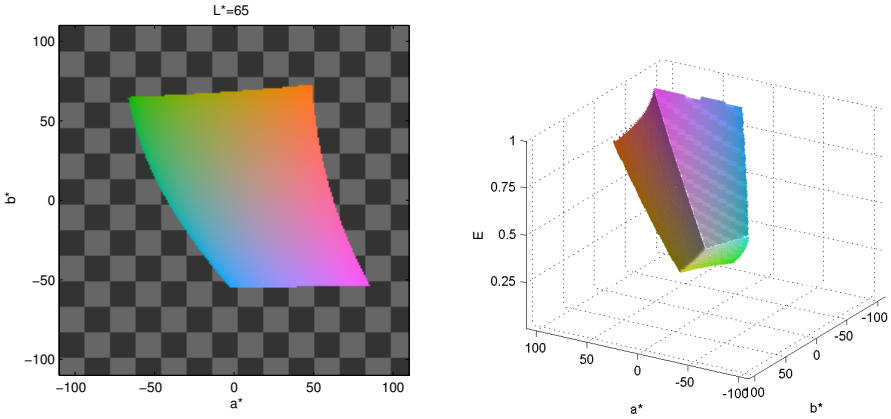
and

$$E_{\text{penalty}}(C_i, C_j) = \begin{cases} 1 - \frac{1}{d} dist(C_i, C_j) & \text{if } dist(C_i, C_j) < d \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

$E_{\max}$ finds among all the colours the maximum linear R, G, or B value (recall that the maximum RGB value is a measure of energy). The terms $C_{i,\text{red}}$, $C_{i,\text{green}}$, and $C_{i,\text{blue}}$ denote the R, G, and B components of colour C_i, respectively. The function E_{penalty} penalizes two colours if they are too close, i.e. less than d apart. The perceptual distance is the Euclidean distance in CIELAB, denoted $dist(C_i, C_j)$. Here, a linear penalty function is used, although nonlinear functions may work as well. The constant k determines the relative importance $E_{\max}$ and E_{penalty}, describing the relative increase of the penalty energy with decreasing colour distance. In our optimization, we let k be equal to the maximum allowable RGB value, or $k = 1$.

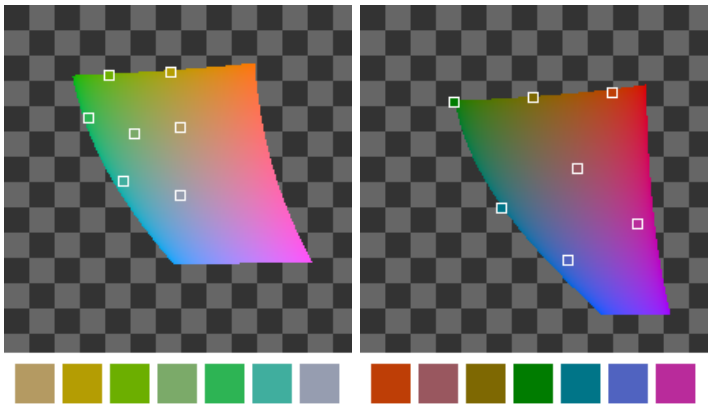
Figure 3.5 illustrates the $L^* = 65$ colour slice from CIELAB, and its energy heightfield. The optimization problem can be thought of as finding N points on the surface of the energy heightfield, such that the 3D Euclidean distance between all points is at least d, and the maximum height among all points is minimized. From Figure 3.5, it can be expected that green and yellow colours use less energy because they are lower in the heightfield. Figure 3.2 shows the same qualitative behavior. Moreover, the energy heightfield and its boundaries, which correspond to the gamut of displayable colours, are quite smooth; thus, optimization is expected to pose no particular difficulties.

The optimization process minimizes E_{cost} with respect to C. Since L^* is fixed, each C_i has two components a_i^* and b_i^* (from the CIELAB representation of C_i), for a total of $2N$ variables. With this choice of variables, the hard constraint of constant L^* is always guaranteed. For optimization, we apply the Nelder and Mead method [31], also known as the downhill simplex method or amoeba method. Other optimization methods for non-linear multi-variable functions may be used alternatively. We apply the optimization method to minimize $E_{\text{cost}}(a_1^*, b_1^*, ..., a_N^*, b_N^*)$. The downhill simplex method works iteratively, starting from an initial configuration. We randomly generate the initial configuration, making sure that all the colours are within gamut. During optimization, colours are also checked to lie within the gamut. If a colour moves outside the gamut, it will be projected back to the closest point on the gamut boundary. In the illustration of Figure 3.5(a), the gamut boundary appears as the boundary between the coloured region and checkerboard background. Finally, to avoid finding only local minima, we run the optimization process several times with



(a) (b)

Figure 3.5: (a) $L^* = 65$ iso-lightness slice from CIELAB. (b) An energy heightfield showing the energy consumption of the colours; the green-yellow region tends to have lower energy. ©Eurographics Association 2009. Reproduced with kind permission of the Eurographics Association.



(a) (b)

Figure 3.6: (a) Optimizing for $N = 7$ colours, iso-lightness level of $L^* = 65$, and minimum colour distance of $d = 30$. (b) Optimizing for $N = 7$, $L^* = 45$, $d = 50$. The top-row images show the a^*-b^* plane, the bottom-row shows the colour sets. ©Eurographics Association 2009. Reproduced with kind permission of the Eurographics Association.

random initializations and keep the best solution. Figure 3.6 shows the result of optimizing for two different input parameters. As expected, optimal colours tend to come from the green and yellow regions, which lead to low energy consumption.

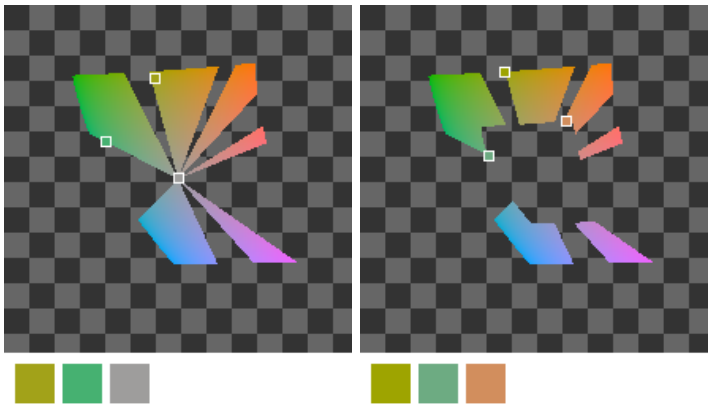
3.3 Hybrid Optimization

So far we have looked at two ways of optimizing for energy aware colours. Both approaches have their advantages and disadvantages. In the discrete optimization, we use colour category as a design constraint so that the resulting colours are distinguishably different in their colour names. We restrict our optimization space to only 6 different, fully-saturated colours. The restrictions on saturation and exact hue angles limit us to a small colour palette (Figure 3.3). On the other hand, the continuous optimization approach allows us to optimize over a larger continuous colour space. There are no constraints on saturation or colour hue, allowing us to achieve a wider range of colours. However, one drawback to the continuous optimization is that the resulting colours can be quite similar. For example, in Figure 3.6(a), there are at least 3 greenish colours. This can be a problem especially when we want to use colour to label data.

We present a third, hybrid approach for energy aware colours. The goal is to allow for colour distinction (which is lacking in the continuous method) and still have the larger colour palette that comes from optimizing over a continuous space (which is lacking in the discrete method). We achieve this through a simple modification of the continuous optimization method. We return to use the six colours from the discrete optimization method, and notice that the six colour hues span a range of angles in LCH_{ab} (recall that LCH_{ab} is the cylindrical transformation of CIELAB [11]). By visual inspection, the 6 colours are roughly bounded between these LCH_{ab} angles: 85° to 105° (yellow), 117° to 153° (green), 47° to 63° (orange), 312° to 325° (purple), 22° to 32° (red), and 226° to 295° (blue). Figure 3.7(a) illustrates these categorical colour subspaces on an iso-lightness slice. As in the continuous method, we optimize with input parameters N, L^*, and d, but we introduce a new parameter S for specifying a minimum colour saturation. For any colour, S is a percentage of the maximum possible chroma. Without the parameter S, grayscale (0 percent saturation) "colours" may appear in our results.

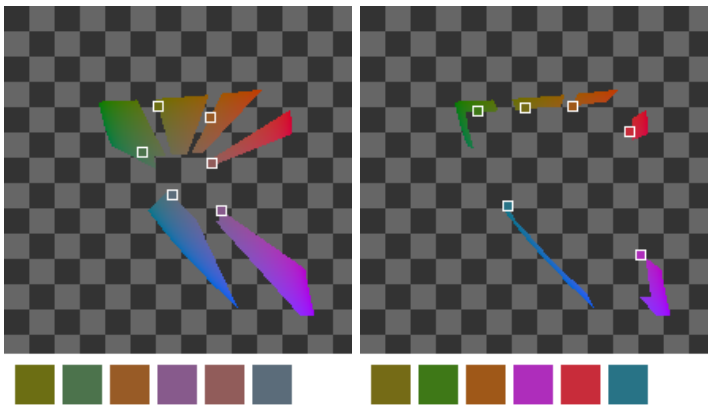
The optimization process is the same as before, except now the initial points must be within the proper angle and saturation ranges, and only one point is assigned to any

categorical colour region. During optimization, if colours move outside their range, a high energy cost is assigned to the overall energy. This will prevent out-of-range colours from being chosen. The optimization process is again run several times with random initializations to avoid getting stuck in a local minima. Figure 3.7 shows an example of optimizing for 3 colours with constrained and unconstrained saturation. Unconstrained saturation basically means that gray colours are allowed in the final colour set. Figure 3.8 shows another example optimizing for 6 colours.



(a) (b)

Figure 3.7: (a) Optimizing for $N = 3$ colours, iso-lightness level of $L^* = 65$, minimum colour distance of $d = 50$, and unconstrained saturation (i.e. gray allowed). (b) Optimizing for $N = 3$, $L^* = 65$, $d = 50$, and minimum saturation level of 50 percent (i.e. no gray).



(a) (b)

Figure 3.8: (a) Optimizing for $N = 6$ colours, iso-lightness level of $L^* = 45$, minimum colour distance of $d = 30$, and at least 30 percent saturation. (b) Optimizing for $N = 6$, $L^* = 45$, $d = 30$, and at least 85 percent saturation.

Chapter 4

Hue-Preserving Blending for Volume Rendering

The volume-rendering integral (Equation 4.1) is commonly used to describe physical light transport in a volume, with optical properties κ (absorption coefficient) and q (source term describing emission) and integration from entry point into the volume, $s = s_0$, to the exit point toward the camera, $s = D$ [10]:

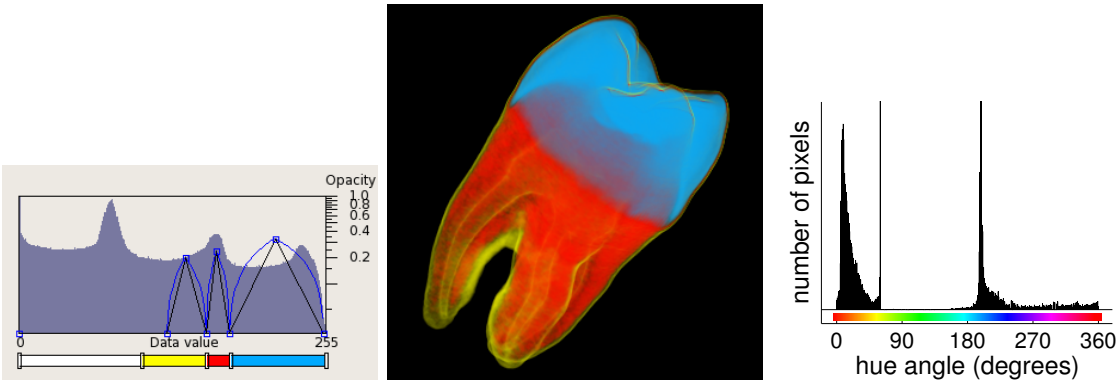
$$I(D) = I_0 e^{-\int_{s_0}^{D} \kappa(t)\,\mathrm{d}t} + \int_{s_0}^{D} q(s) e^{-\int_{s}^{D} \kappa(t)\,\mathrm{d}t}\,\mathrm{d}s \tag{4.1}$$

The discretized version of Equation 4.1 can be solved for each ray passing through the volume using either a front-to-back or back-to-front compositing scheme [10] (Equations 4.2 and 4.3, respectively) .

$$\begin{aligned} C_{\mathrm{dst}} &\leftarrow C_{\mathrm{dst}} + (1 - \alpha_{\mathrm{dst}}) C_{\mathrm{src}} \\ \alpha_{\mathrm{dst}} &\leftarrow \alpha_{\mathrm{dst}} + (1 - \alpha_{\mathrm{dst}}) \alpha_{\mathrm{src}} \end{aligned} \tag{4.2}$$

$$C_{\mathrm{dst}} \leftarrow (1 - \alpha_{\mathrm{src}}) C_{\mathrm{dst}} + C_{\mathrm{src}} \tag{4.3}$$

In general, the emission and absorption coefficients of Equation 4.1 are unknown, and it is up to the user to assign these optical properties through a *transfer function*. A transfer function maps optical properties such as colour to abstract volume scalar data. In Figure 4.1, we show a typical transfer function for the tooth data set and the resulting image generated



(a) (b) (c)

Figure 4.1: (a) A transfer function assigns optical properties in the form of colour and opacity to scalar data. (b) A rendering of the tooth data set using general raycasting, and (c) its colour hue histogram showing the presence of extraneous hues that are not specified in the original transfer function.

using general raycasting. Although the transfer function in Figure 4.1(a) only specifies 3 colour hues, the rendered tooth image of Figure 4.1(b) contains additional hues, as verified by the colour hue histogram in Figure 4.1(c).

In any kind of visualization, it is desirable to have colours that are easy to distinguish and identify. Colour is frequently used for visual grouping and labeling, which is most effective by means of chromatic information, as opposed to luminance information [19, 43]. Colour is also frequently employed in volume rendering to identify differently classified materials. However, in volume rendering, traditional colour blending during the compositing stage produces unpredictable, off-colour hues that are not specified in the transfer function. For example, in Figure 4.1(b) we see tints of orange and purple even though the transfer function only specifies yellow, red, and blue. In this chapter, we present a new colour blending method that aims to preserve colour hue.

Most relevant for our work is the recent publication by Wang et al. [41]. They investigate and provide guidelines and rules for colour design for illustrative visualization. In particular, they describe the appropriate choice of colours for semitransparent layers: colours should have opposite hue in order to avoid hue shift after blending. In the case of more than two semitransparent layers, they propose further constraints on the input colours. One of their guideline variants is to assign two colours with opposite hues for the two most important image elements and a more neutral colour for the less important element(s).

An alternative guideline is to change the input colours locally: they recommend reducing the saturation of the background element in overlap regions. The (geometric) overlap is detected by depth peeling. We adopt the very idea that hue shift should be avoided, but guarantee hue preservation by a generic blending model that allows for arbitrary number and configuration of input colours, rather than being restricted to only certain pairs of opposite colours. In particular, we provide a complete computational and parameter-free model that may be applied to any kind of compositing problem and without constraints on the colour maps.

4.1 Design of Hue-Preserving Blending

We discuss the perceptual motivation and the design considerations for the development of hue-preserving blending before we present the respective computational model in the next section. Since we target perceptual transparency, our compositing approach is not subject to any physical constraints, but can be formulated as an algebraic model. The discussion is initially restricted to compositing two overlaid images, and it will be later extended to compositing several images and even to continuous compositing in volume rendering. The primary goal of the new compositing model is to support easy perception of distinct colours for labeling, in combination with a good perception of transparent overlays.

There is strong and ample empirical evidence that image luminance has the most impact on transparency perception. In fact, most studies have focused on investigating achromatic configurations; see, for example [25, 2, 13, 20]. While the crucial role of the achromatic channel is undisputed for perceptual transparency, a large portion of the perception literature indicates that chromatic information has very limited influence on transparency perception. For example, Nakayama et al. [30] report that transparency perception is robust under a wide range of colour configurations, both for the occluder and the occludee. Similarly, Anderson [1] identifies achromatic contrast as the primary determinant of scission. An extreme view would remove chromatic information completely from a transparency model. Such a view is quite accepted for the perception of motion, where chromatic contrast apparently plays (almost) no role; see, for example [33, 24]. However, there is also some evidence that special configurations of chromatic contrast alone can trigger transparency perception [9]. For example, the colour of the overlay image should share hue properties with the images underneath [4]. As a consequence of the unclear role of chromatic information, we favor

a conservative approach to perceptual transparency by focusing on well accepted models of luminance composition and by reducing the impact of the chromatic channels. In particular, we favour complete preservation of hue. Summarizing previous work in perceptual psychology and psychophysics, the following observations can be made:

[O1] Perceptual research indicates that luminance is most important for the perception of transparency.

[O2] Shape perception by shape-from-shading is based on luminance information.

[O3] The chromatic channels play a major role in visual grouping; hue is particularly well suited for visual labeling, e.g., of nominal data.

[O4] Chromatic information and especially saturation play a minor—at least unclear—role for transparency perception.

From these observations, we derive at the following design criteria:

[D1] Any new compositing model has to exhibit the same behavior for the luminance channel as established compositing models. According to *[O1]*, luminance is critical for transparency perception, and there exist models with demonstrated effectiveness. In addition, the achromatic channel may carry important information, such as shape-from-shading information *[O2]*, that should not be interfered with.

[D2] The same, constant hue should be used for each nominal data entry to facilitate visual grouping *[O3]*.

[D3] Artificial colour discontinuities should be avoided for continuously varying input colours, so that artificial perceptual contours are avoided.

These design criteria guide the construction of a generalized compositing operator. According to Porter and Duff [32], a wide range of compositing strategies can be formulated as the weighted sum of two colours. In particular, their approach includes alpha blending (the *over* operator), typically used for computing transparent overlays. We adopt the compositing idea by Porter and Duff and add just a little modification: instead of a direct, component-wise sum of two colours C_1 and C_2, a new "add" operator is proposed that meets the above design criteria. We denote traditional addition of colours by the symbol "+" and

the new operator by "$\oplus$". In this notation, the hue-preserving sum of colours is:

$$C_{\text{new}} = C_1 \oplus C_2 \tag{4.4}$$

From the above design criteria, we impose the following requirements that hue-preserving colour addition has to meet:

[R1] The same luminance behavior as in traditional summation for the achromatic case should be achieved: the luminance $(C_1 \oplus C_2)$ should be identical to the sum of the luminances of C_1 and C_2.

[R2] $Hue(C_{\text{new}}) \in \{Hue(C_1), Hue(C_2)\}$. The hue of C_{new} is either equal to the hue of C_1 or C_2. The hue of C_{new} is chosen as the hue of the dominating of the two colours C_1 and C_2. The dominating colour is the one whose hue would be closest to the blended colour in traditional colour summation.

[R3] Saturation variations are used to avoid colour discontinuities. When the dominating colour, and thus the final hue, is to change, C_{new} should go through the gray point with vanishing saturation, so that even an abrupt change of hue does not imply a discontinuity in chromaticity.

The requirements *[R1]* and *[R3]* correspond directly to the design criteria *[D1]* and *[D3]*. However, the design criterion *[D2]* cannot be implemented completely because it asks for conflicting choices of hue: if two different nominal data entries are composited, not both of their hues can survive. The requirement *[R2]* approximates *[D2]* by choosing the dominant hue.

The semantics and mathematical structure of the new $\oplus$ operator is designed to resemble the traditional $+$ operator as much as possible, so that it can be used in any existing blending algorithms, especially in compositing schemes for volume rendering. The $\oplus$ operator is binary: it takes two input colours. The extension to compositing several image layers or to many samples along viewing rays in volume rendering is possible by applying $\oplus$ several times along the image compositing stack. The mechanics and mathematical definition of the $\oplus$ operator are presented in the following section.

4.2 Mechanics of Hue-Preserving Blending

This section presents the computational model of hue-preserving blending that follows the requirements *[R1]–[R3]*. We aim at a generic compositing model, modifying the Porter and Duff image compositing approach. In its original form, any Porter and Duff operator can be written as a weighted sum of two input colours C_A and C_B [32]:

$$(\alpha_A F_A)C_A + (\alpha_B F_B)C_B \tag{4.5}$$

where α_A and α_B are the alpha values associated with the two colours and F_A and F_B are respective fractional components. The scalar values $(\alpha_A F_A)$ and $(\alpha_B F_B)$ can be interpreted as combined weights for the two input colours. The original version of those compositing operators assumes colours in RGB colour space. However, any other colour space related to RGB by linear transformation may be employed, e.g. CIE XYZ. The basis of colour computation is the tristimulus theory, which interprets colour as elements in a 3D vector space.

Equation (4.5) contains two relevant arithmetic operations: the multiplication of a scalar weight with a 3D colour, and the sum of 3D colours. With hue-preserving blending, multiplication with a scalar weight remains unchanged. The only difference is that the traditional component-wise addition by the + operator is replaced by the new operator $\oplus$ from Eq. (4.4).

The hue-preserving $\oplus$ operator is based on computations in a set of appropriate colour representations: in hue, saturation, and brightness components that are modified separately. As discussed before, there exist several different colour spaces with such a split, e.g., HSL, HSV, or LCH_{ab}. We have chosen HSL because it is widely used in colour pickers, it has easy transformation rules to and from RGB, and its lightness axis conforms more to perceived brightness than HSV. In the following, we follow the usual practice of basing HSL directly on sRGB. Please note that the idea of hue-preserving colour blending may be reformulated in any other colour space that provides measures for hue, saturation, and brightness.

Figure 4.2 illustrates and compares traditional blending with hue-preserving blending. Figure 4.2(a) sketches the geometry of blending in the hue–saturation plane—with hue as angle and saturation as radial distance from the center. The two exemplary input colours, teal and orange, are marked by small white circles. Depending on the relative weights assigned to the two colours, the result of traditional blending yields a colour on the long dashed line crossing several colour hues. The possible resulting hues are also shown in the

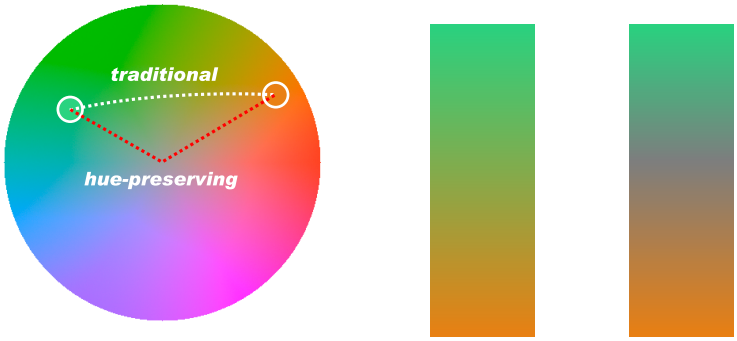


Figure 4.2: (a) Traditional blending of two colours yields various colour hues (indicated by long dashed lines). In contrast, hue-preserving colour blending mixes the two colours so that they go through the gray point (short dotted line), avoiding any extraneous hues. (b) Traditional alpha blending of teal and orange. (c) Hue-preserving alpha blending of teal and orange. Note the presence or absence of the yellowish hue in both colour profiles.

colour bar in Figure 4.2(b). Our aim is to modify the traditional blending + operator so that when two colours are blended, the resulting colour only has the same hue as either of the original ones, as shown in Figure 4.2(c). The basic idea is to blend two colours through the middle gray point (or the central axis, where colour saturation equals zero), as illustrated by the dotted line in Figure 4.2(a).

In other words, hue-preserving blending can be essentially split in two pieces: blending from one input colour C_1 towards the gray axis (which keeps the hue of C_1), or blending from the other input colour C_2 towards the gray axis (which keeps the hue of C_2). We decide which of the two pieces is used by examining the relative "strengths" of the two input colours; the hue of the dominant colour determines the hue of the blended colour. The dominant colour is computed by first blending C_1 and C_2 temporarily using the traditional + operation to produce C_{trad}. Its hue, H_{trad}, is then compared with the hue of C_1 and C_2, H_1 and H_2 respectively; and if H_{trad} is closer to H_1, then C_1 is dominant, otherwise C_2 is dominant. The actual compositing step has to ensure that the dominant hue does not change. This is achieved by modifying the non-dominant colour in a way that it becomes the opposite of the dominant colour; the saturation and lightness of the non-dominant colour stay the same. By adding opposite colours, the colour moves towards the gray point, and we guarantee that the original hue does not change. Figure 4.3 illustrates this idea. Finally, the lightness component of the blended colour is determined by adding the lightness components

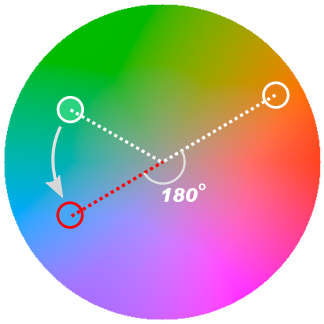


Figure 4.3: Blending opposite (i.e., complementary) colours in the traditional colour blending model leads to a more neutral colour and preserves either original hue. We follow the same idea in our hue-preserving colour blending model. Given two arbitrary colours (circled in white) that are not necessarily opposite to each other, we modify only the hue component of one of the colours to be the opposite hue of the other colour (circled in red), then they are added as before. Choosing which colour to modify the hue depends on a comparison with the resulting hue from traditional blending.

of the two input colours: $L_1 + L_2$. This separate compositing of lightness implements *[R1]* of Chapter 4.1.

Algorithm 1 describes our hue-preserving blending model. Note that when it comes to colour hue comparison, we treat a gray colour as any other colour, but with zero saturation. Therefore, gray and any other colour are considered to have the same hue. Also note that if two input colours have the same hue, our result is identical to traditional blending.

The hue-preserving blending method described so far is non-associative. The resulting blended colour is different whether colour samples along a ray are composited in front-to-back or back-to-front order. One possible reason for the non-associative nature may be the different behaviour between the front-to-back and back-to-front compositing equations (see Equations 4.2 and 4.3). Although the source colours are the same in either direction of a ray (but reversed), the intermediary destination colour values of front-to-back and back-to-front blending are different, causing the final blended colour to look different. A comparison between front-to-back and back-to-front blending is shown in the following section.

Algorithm 1 Calculating $C_{\text{new}} = C_1 \oplus C_2$

Require: C_1 and C_2 are valid RGB colours
Ensure: C_{new} is a valid RGB colour
  $C_{\text{trad}} = C_1 + C_2$
  $HSL_1 = RGB2HSL(C_1)$
  $HSL_2 = RGB2HSL(C_2)$
  $HSL_{\text{trad}} = RGB2HSL(C_{\text{trad}})$
  if $H_1 = H_2$ **then**
    $C_{\text{new}} = C_{\text{trad}}$
  else $\{H_1 \neq H_2\}$
    if H_{trad} closer to H_1 **then**
      $H'_1 = H_1 + 180°$
      $C'_2 = HSL2RGB(H'_1, S_2, L_2)$
      $C_{\text{new}} = C_1 + C'_2$
    else $\{H_{\text{trad}}$ closer to $H_2\}$
      $H'_2 = H_2 + 180°$
      $C'_1 = HSL2RGB(H'_2, S_1, L_1)$
      $C_{\text{new}} = C'_1 + C_2$
    end if
    {lightness computation}
    $HSL_{\text{temp}} = RGB2HSL(C_{\text{new}})$
    $C_{\text{new}} = HSL2RGB(H_{\text{temp}}, S_{\text{temp}}, L_1 + L_2)$
  end if

4.3 Examples of Hue-Preserving Blending

We illustrate the effects of hue-preserving blending for several different examples of image compositing, and compare them to traditional blending. First, we start with the simple case of alpha blending two colours in a hue-preserving way. Two colours C_1 and C_2 are alpha-blended:

$$C_{\text{new}} = (1 - \alpha)C_1 \oplus \alpha C_2 \tag{4.6}$$

Figure 4.4 compares pairs of alpha-blended colour profiles using traditional and hue-preserving blending. The two input colours are at opposite ends of each colour profile, and alpha ranges from 0 to 1. It is easy to see that the hue-preserving blending produces no extra hues other than the original ones. A nice property of our method is that blending opposite colours or blending same-hue colours yields the same result as traditional blending, as shown in Figure 4.4(c)–Figure 4.4(e).

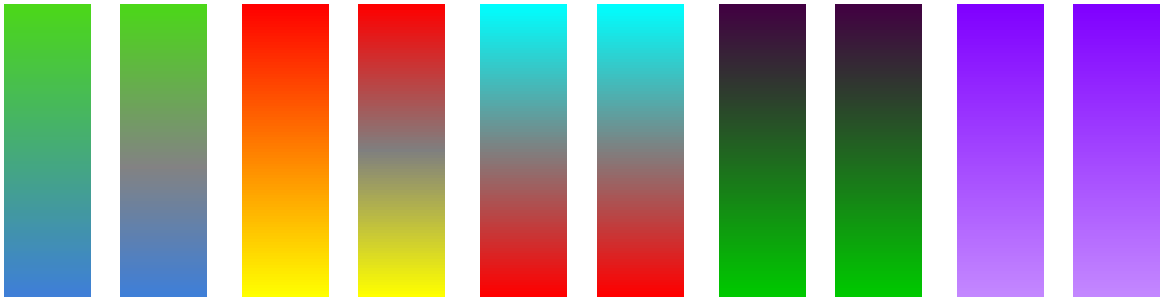


Figure 4.4: In each pair, traditional (left) and hue-preserving (right) alpha blending for two colours are compared side by side. Images (a) and (b) show the typical cases where hue-preserving blending employs colour transitions through gray to avoid extraneous hues. Images (c) and (d) show that for blending opposite colours our method gives the same result as traditional blending. Image (e) demonstrates blending two colours of the same hue, which also yields the same result as traditional blending.

Next, we extend the alpha blending of only two colours to the more complex example of blending several colours normally encountered in volume rendering. In volume rendering applications, it is typical for users to choose a few distinct colours for visual labeling of classified materials during data exploration (usually 1–6 material colours). However, as the number of chosen colours exceeds 1, the colours that can result from traditional blending cover a large and continuous range of different hues. Figure 4.5 compares the possible

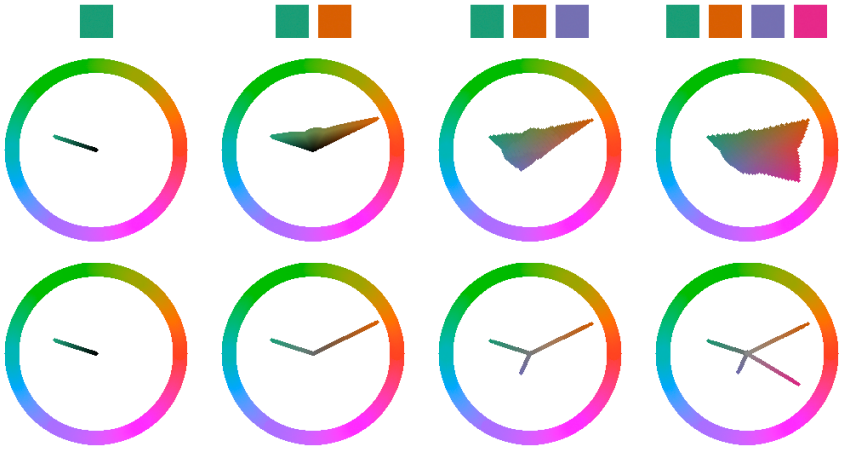


Figure 4.5: In volume rendering, many colours of various hues may be mixed. At the top, we show the colours used in the blending. The next (middle) row shows all possible colours that can result using traditional colour blending. The last (bottom) row illustrates those colours that originate from hue-preserving blending. Note that the possible colours are viewed in the HSL colour cone from above (showing hue by angle and saturation by radius), so that the lower-lightness colours are occluded. We surround all possible fully saturated colours in HSL colour circles to aid readers in identifying colour hues.

colours that can result from blending up to 4 colours in both the traditional (middle) and hue-preserving (bottom) methods. The colours are displayed in their respective coordinates in the HSL double-cone, viewed from above (i.e., looking down the HSL double cone from where $L = 1.0$): the traditional approach covers a large portion of the HSL colour space as many mixed colours are introduced, whereas the hue-preserving approach is limited to its distinct input hues.

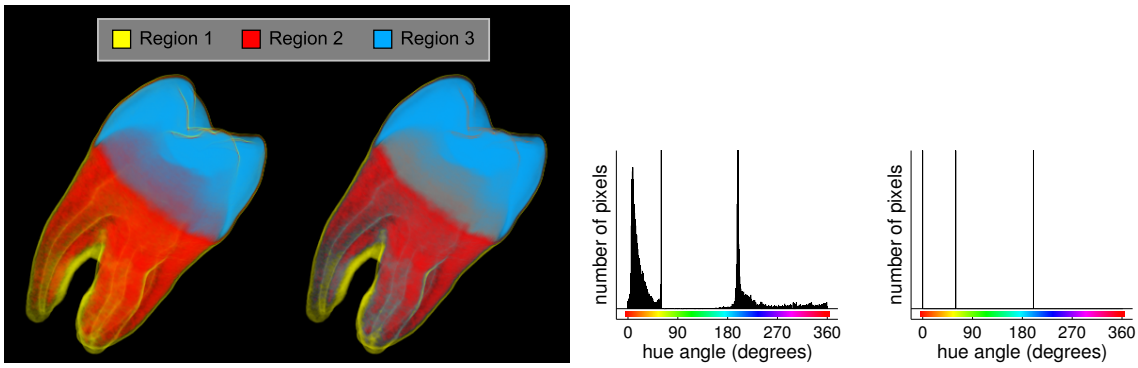
We now apply our blending technique to actual volume visualization of 3D scalar data sets. Images are rendered by front-to-back raycasting with optical properties such as colour and opacity assigned to data values via a 1D transfer function. Figure 4.6 compares traditional and hue-preserving rendering of the tooth data set seen earlier in Figure 4.1. In the traditional blending, the 3 input colours yellow, red, and blue mix to produce tints of orange and purple. The presence of these off-colour hues is quantitatively documented in the colour hue histogram in Figure 4.6(b). Using the new blending method, only the original 3 colours are present, as shown by the hue histogram in Figure 4.6(c). In this way, colour labeling is improved at no loss of feature identification.

Figure 4.7 shows the volume rendering of a human chest data set. In Figure 4.7(a),

we use opposite colours blue and yellow, and show that our approach produces the same result as traditional blending. However, when the blue flesh colour moves its hue toward cyan, the traditional blending produces an undesirable tint of green, whereas our approach does not. This example was designed to resemble the colour choice by Wang et al. [41] in their Figure 8. If opposite colours are chosen according to their guidelines, hue-preserving blending is identical to traditional blending. However, we have essentially given the user the freedom to select arbitrary colours without having to worry about generating extraneous hues and false, mixed colours.

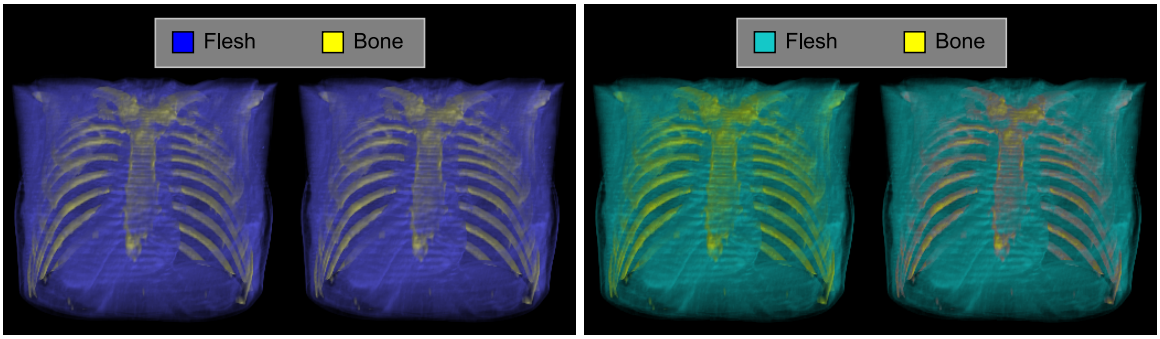
The smooth transition of colours through gray, as required by *[R3]*, is demonstrated in Figure 4.8. Here, the opacity of the brain is gradually increased (from left to right). With increasing opacity, that inner part of the volume data set is becoming more and more pronounced and the respective colour (red) is increasingly more dominant. The transition from dominant exterior colour (green) and dominant interior colour goes through gray with smooth variations of saturation.

Finally, Figure 4.9 compares the traditional and hue-preserving renderings of the segmented frog data set with 5 different colour labels. This example demonstrates that hue-preserving blending can work with a larger number of input hues. Additional comparisons of traditional and hue-preserving blending for different data sets are shown in Figure 4.10 and Figure 4.11. Figure 4.11 (middle and right) shows that the non-associative nature of hue-preserving blending (as described in Chapter 4.2) give different results when a volume is rendered front-to-back or back-to-front.



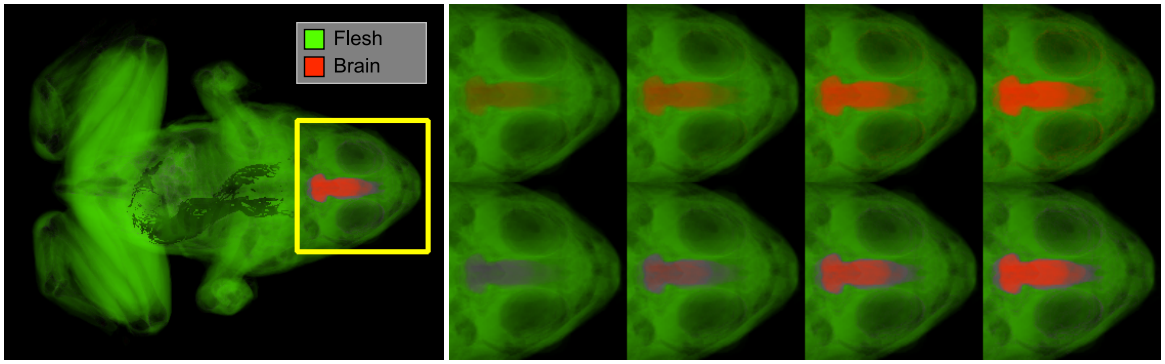
(a) (b) (c)

Figure 4.6: (a) Traditional (left) and hue-preserving (right) rendering of the tooth data set. In the traditional rendering, orange colours can be seen where red and yellow mix. There are also purple hues where red and blue mix. These extraneous hues completely disappear in the hue-preserving rendering. The colour hue histograms for both renderings are shown in (b) and (c). Note the three vertical lines in the hue-preserving histogram, representing the original colour hues.



(a) (b)

Figure 4.7: (a) Traditional (left) and hue-preserving (right) rendering of a chest data set, using opposite colours blue and yellow. Since the original colours are already opposite to each other, the traditional method does not suffer from extraneous hues, and in fact looks just like the hue-preserving rendering. (b) The blue hue of the flesh is offset toward cyan, and we immediately see that traditional blending produces tints of green. This, however, does not pose a problem for hue-preserving blending, which still maintains only cyan and yellow.



(a) (b)

Figure 4.8: (a) Volume rendering of a segmented frog data set with only the flesh and brain shown. (b) We illustrate the effect of increasing the brain opacity (left to right) in both the traditional (top) and hue-preserving (bottom) methods. The gray colours in the hue-preserving approach indicate the smooth transitions between the two colours.

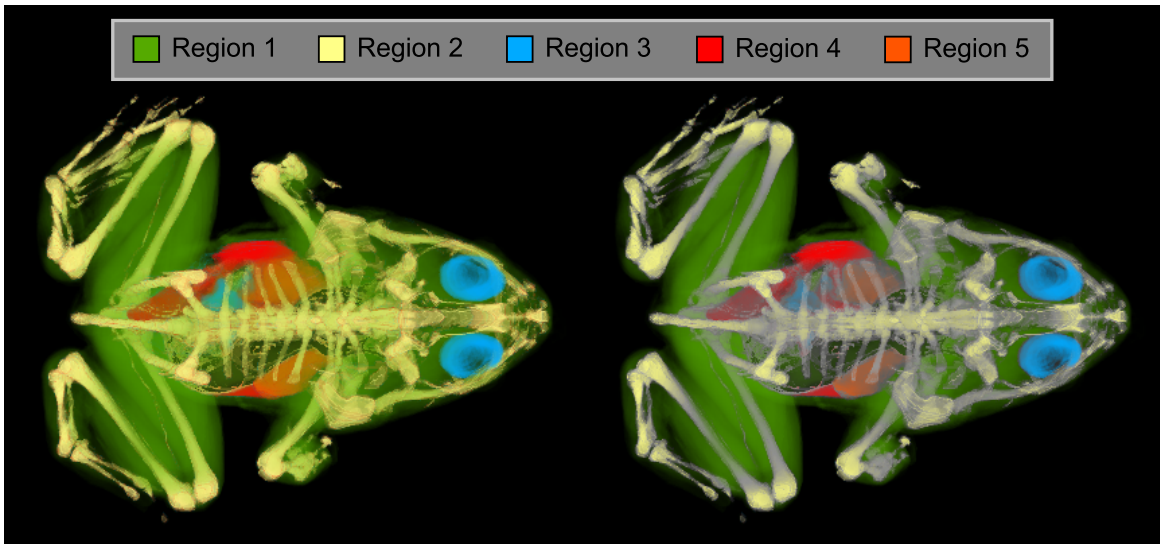


Figure 4.9: Another comparison between the traditional (left) and hue-preserving (right) rendering of the segmented frog data set, with more features identified and colour-coded. In traditional colour blending, the colour of the bone structure shifts into a vibrant, highly saturated, yellow (caused by mixing with surrounding green). In addition, the boundaries between the inner red and orange organs are unclear. In contrast, hue-preserving blending is not subject to those colour shifts.

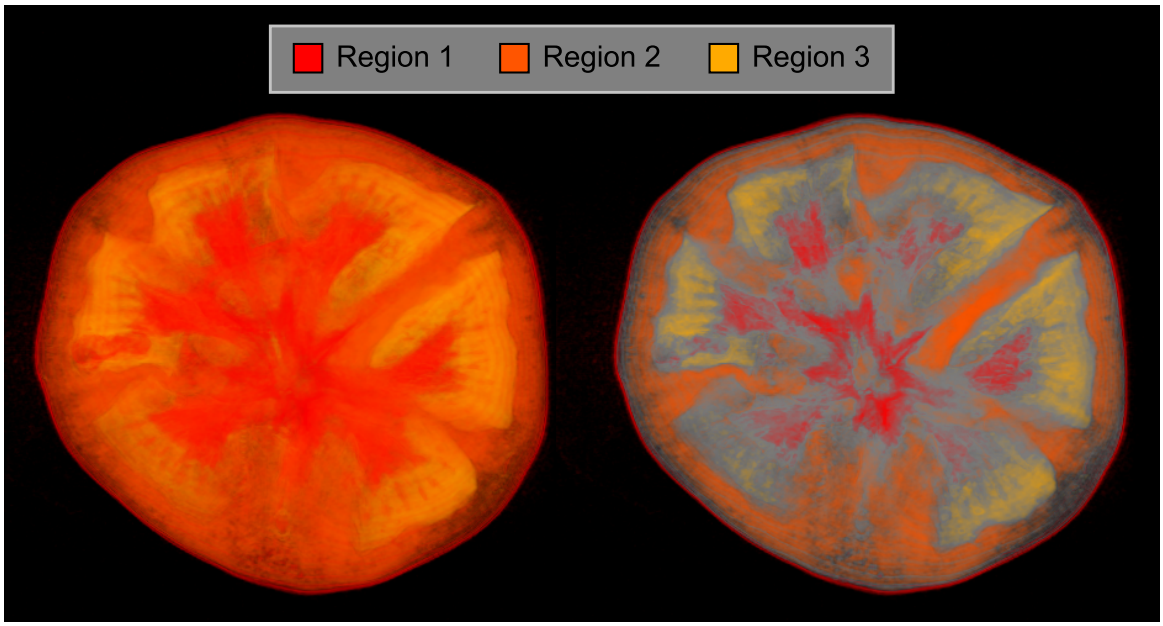


Figure 4.10: Traditional (left) and hue-preserving (right) rendering of a tomato data set.

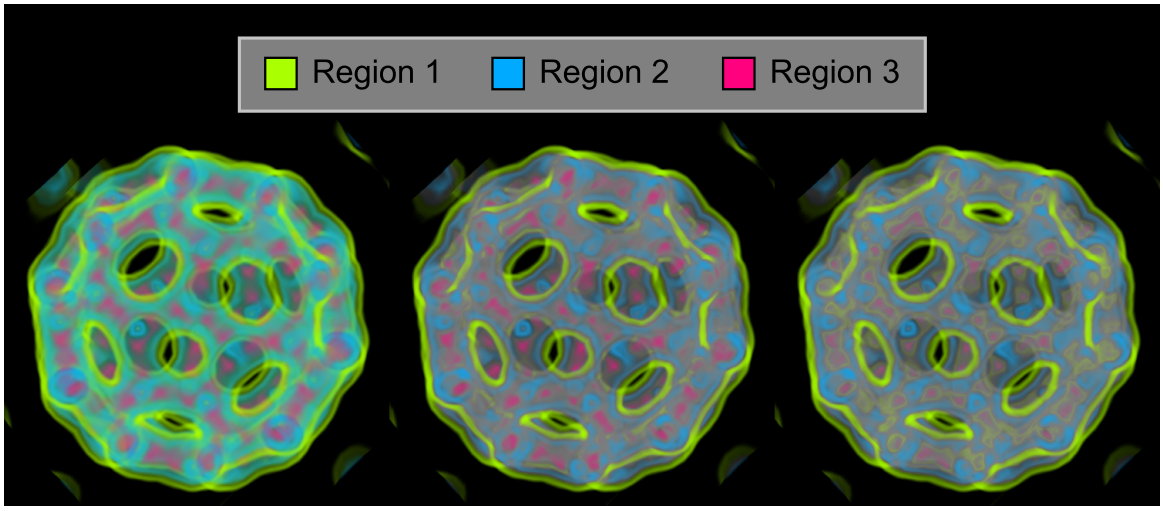


Figure 4.11: Traditional (left), hue-preserving front-to-back (middle), and hue-preserving back-to-front (right) rendering of the bucky ball data set. The colour compositing order affects the final blended result.

Chapter 5

Results and Discussion

Sets of discrete colours may be used in various visualization applications that aim at showing nominal data or membership and grouping information. In this chapter, we present two typical examples. The first example is a 2D mapping that may appear in cartography, geospatial information systems, or other 2D data visualization. The second example is from direct volume visualization, where discrete colours are employed to identify differently classified materials. We intend to demonstrate that energy aware colours can be chosen for such typical visualization applications without sacrificing perceptual distinguishability.

5.1 Energy Aware Colours for Visualization of 2D Data

Figure 5.1 shows an example of a 2D map with discrete colours. The map is taken from ColorBrewer [14]. We have chosen ColorBrewer as a ground for comparison because ColorBrewer is specifically designed for creating colour mappings for 2D maps, facilitating good visual perception. The map in Figure 5.1 uses the *4-class qualitative Dark2* colouring scheme, which contains four near-isoluminant colours. A summary of the colours is listed in Table 5.1. The colour details, such as average lightness, minimum colour saturation, and minimum colour distance are used in our subsequent optimizations to find low energy colour sets.

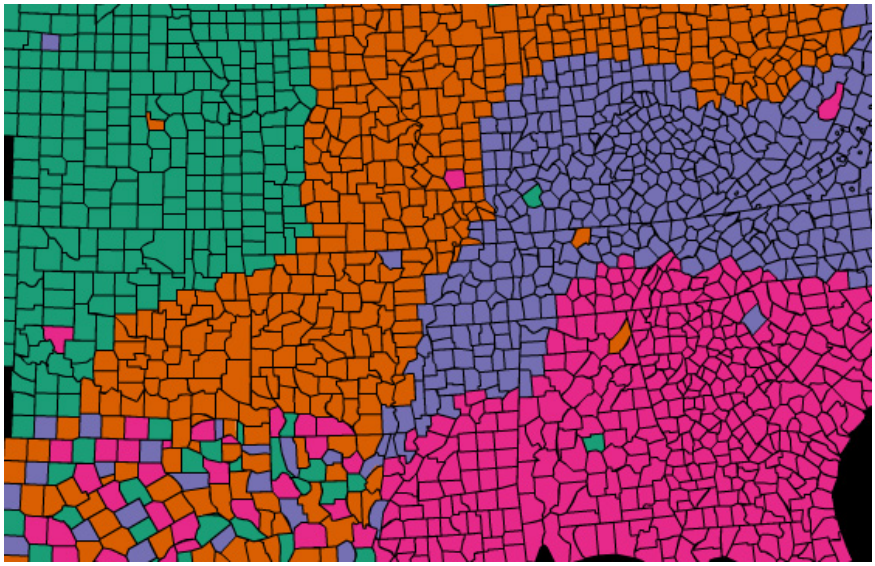


Figure 5.1: A colour map using ColorBrewer's *4-class qualitative Dark2* colouring scheme [14]. ©Eurographics Association 2009. Reproduced with kind permission of the Eurographics Association.

	teal	orange	purple	magenta
R	27	217	117	231
G	158	95	112	41
B	119	2	179	138
L^*	58.05	55.01	50.24	52.29
a^*	-43.05	44.34	18.04	75.05
b^*	10.97	63.67	-34.91	-6.27
saturation	0.95	0.99	0.45	0.91
Average L^*: 53.90 Minimum $L^*a^*b^*$ colour distance: 63.80 Maximum $L^*a^*b^*$ colour distance: 119.36				

Table 5.1: Colour summary of ColorBrewer's *4-class qualitative Dark2* colouring scheme.

In our first approach using a lookup table for discrete iso-lightness colours (Chapter 3.1, Figure 3.3), we only need to specify the lightness. We take the average CIELAB lightness of the 4 ColorBrewer colours, $L^* = 53.90$, to look up 4 energy aware colours in our discrete optimization result. Figure 3.3 recommends picking yellow, green, orange, and purple in that order. A summary of those colours is listed in Table 5.2. There are 4 factorial ways to replace the old colours with the new energy aware colours, so we choose the colour assignment that is most perceptually similar to the original. We apply the colours from the first discrete optimization to the map to obtain Figure 5.4(b).

In our second approach (Chapter 3.2) of optimizing in the continuous CIELAB space, we optimize for 4 colours using the input parameters $N = 4$, $L^* = 53.90$, and $d = 63.80$. This translates to finding 4 low energy colours on the iso-lightness $L^* = 53.90$ CIELAB slice, such that their minimum distance is at least $d = 63.80$. The minimum distance $d = 63.80$ is chosen identical to the smallest CIELAB colour distance between the original 4 ColorBrewer colours (see Table 5.1). The result of the continuous optimization is shown in Figure 5.2, and a summary of the colours is listed in Table 5.3. The resulting colours are applied to the map to obtain Figure 5.4(c).

In our third approach (Chapter 3.3) using hybrid optimization, we optimize for 2 different sets of 4 colours. Depending on whether grayscale colours are preferred, there can be 2 separate colour sets. If we do not allow for grayscale colours (i.e. there is a saturation constraint), the optimization uses the input parameters $N = 4$, $L^* = 53.90$, $d = 63.80$, and $S = 0.45$. The only difference if we allow for grayscale colours (i.e. unconstrained saturation) is $S = 0$, which means that the minimum saturation can be as low as 0 percent. In the optimization that does not allow for gray, we choose $S = 0.45$ which is identical to the minimum saturation of the original 4 ColorBrewer colours (see Table 5.1). The results from the optimization are shown in Figure 5.3 and a summary of the colours is listed in Tables 5.4 and 5.5. The resulting colours are applied to the map to obtain Figures 5.4(d) and 5.4(e).

From visual inspection, the five images of Figure 5.4 provide visual encoding and distinguishability on similar levels. Furthermore, Figure 5.4(c), 5.4(d), and 5.4(e) guarantees the same CIELAB distinguishability as the original colour map from Figure 5.4(a); i.e., based on CIELAB, these four colour maps provide the same quality of perceptual distinguishability.

Figure 5.5 documents the energy consumption of the five images of Figure 5.4. Energy consumption is measured according to our energy model from Equation 2.1, using various

grid sizes. The most prominent observation is the substantial energy savings achieved by colours chosen according to our continuous optimization approach. A typical number of tiles for practical applications is in the range of 500 to 1000; for comparison, 760 LEDs (i.e., tiles) are reported for the HDR display developed by Seetzen et al. [36]. On OLED displays with one or more LEDs per pixel, a higher number of tiles can be used. The number of tiles corresponding to 1 (on the far left of the energy plot) can be thought of as a current main stream display using a single backlight that can be dimmed. For a practical display using 760 tiles, we save up to 44 percent in energy by using colours from the continuous optimization versus ColorBrewer colours. In the discrete optimization case, we see an increase in energy when the grid resolution is low (less than 288 tiles). This is due to our magenta colour having a large RGB value. Both hybrid optimization approaches also result in lower energy cost compared to the original colours. The hybrid method actually uses less energy when grayscale colours are allowed. This is expected, because in an iso-lightness colour slice, grayscale colours tend to have lower energy. In general, the energy consumption drops with increasing number of tiles because the energy control is becoming more fine-grained.

	green	orange	yellow	magenta
R	0	203	134	218
G	151	101	134	0
B	0	0	0	252
L^*	54.16	54.03	54.15	53.95
a^*	-58.28	35.84	-13.36	92.44
b^*	56.26	62.39	60.08	-69.76
saturation	1.00	1.00	1.00	1.00
Average L^*: 54.07 Minimum $L^*a^*b^*$ colour distance: 44.99 Maximum $L^*a^*b^*$ colour distance: 196.47				

Table 5.2: Colour summary from the discrete optimization for the 2D ColorBrewer map.

	green	brown	blue	purple
R	44	166	0	166
G	148	120	144	110
B	14	51	140	161
L^*	53.75	53.79	53.85	53.84
a^*	-51.88	10.94	-32.65	30.59
b^*	53.80	43.71	-7.13	-18.48
saturation	0.96	0.72	1.00	0.34
Average L^*: 53.81 Minimum $L^*a^*b^*$ colour distance: 63.63 Maximum $L^*a^*b^*$ colour distance: 109.67				

Table 5.3: Colour summary from the continuous optimization for the 2D ColorBrewer map.

	green	orange	brown	purple
R	0	221	142	174
G	150	85	128	102
B	0	1	82	182
L^*	53.82	53.89	53.81	53.89
a^*	-58.00	50.43	-1.55	41.62
b^*	55.99	63.60	26.89	-30.50
saturation	1.00	1.00	0.46	0.45
Average L^*: 53.85				
Minimum $L^*a^*b^*$ colour distance: 63.51				
Maximum $L^*a^*b^*$ colour distance: 131.92				

Table 5.4: Colour summary from the hybrid optimization (with constrained saturation) for the 2D ColorBrewer map.

	green	orange	gray	purple
R	80	197	130	182
G	144	104	129	95
B	1	2	120	189
L^*	53.66	53.79	53.81	53.81
a^*	-40.74	31.84	-1.32	48.97
b^*	56.45	61.39	5.00	-34.59
saturation	1.00	0.99	0.086	0.54
Average L^*: 53.77				
Minimum $L^*a^*b^*$ colour distance: 64.01				
Maximum $L^*a^*b^*$ colour distance: 127.82				

Table 5.5: Colour summary from the hybrid optimization (with unconstrained saturation) for the 2D ColorBrewer map.

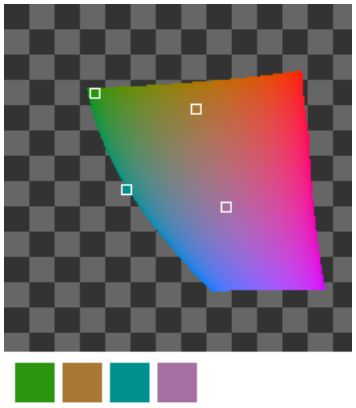


Figure 5.2: Result from the continuous optimization using input parameters $N = 4$, $L^* = 53.90$, and $d = 63.80$.

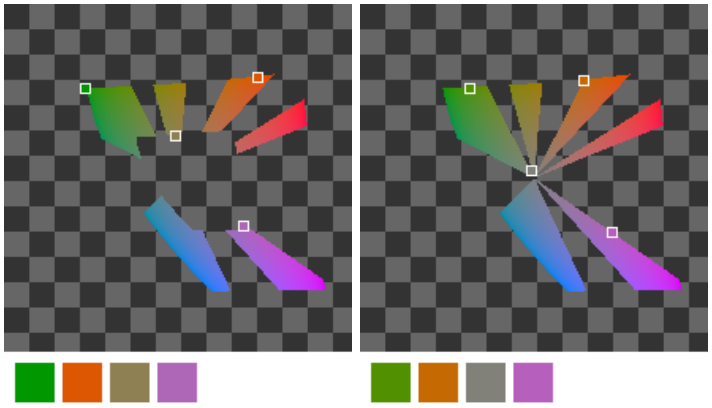
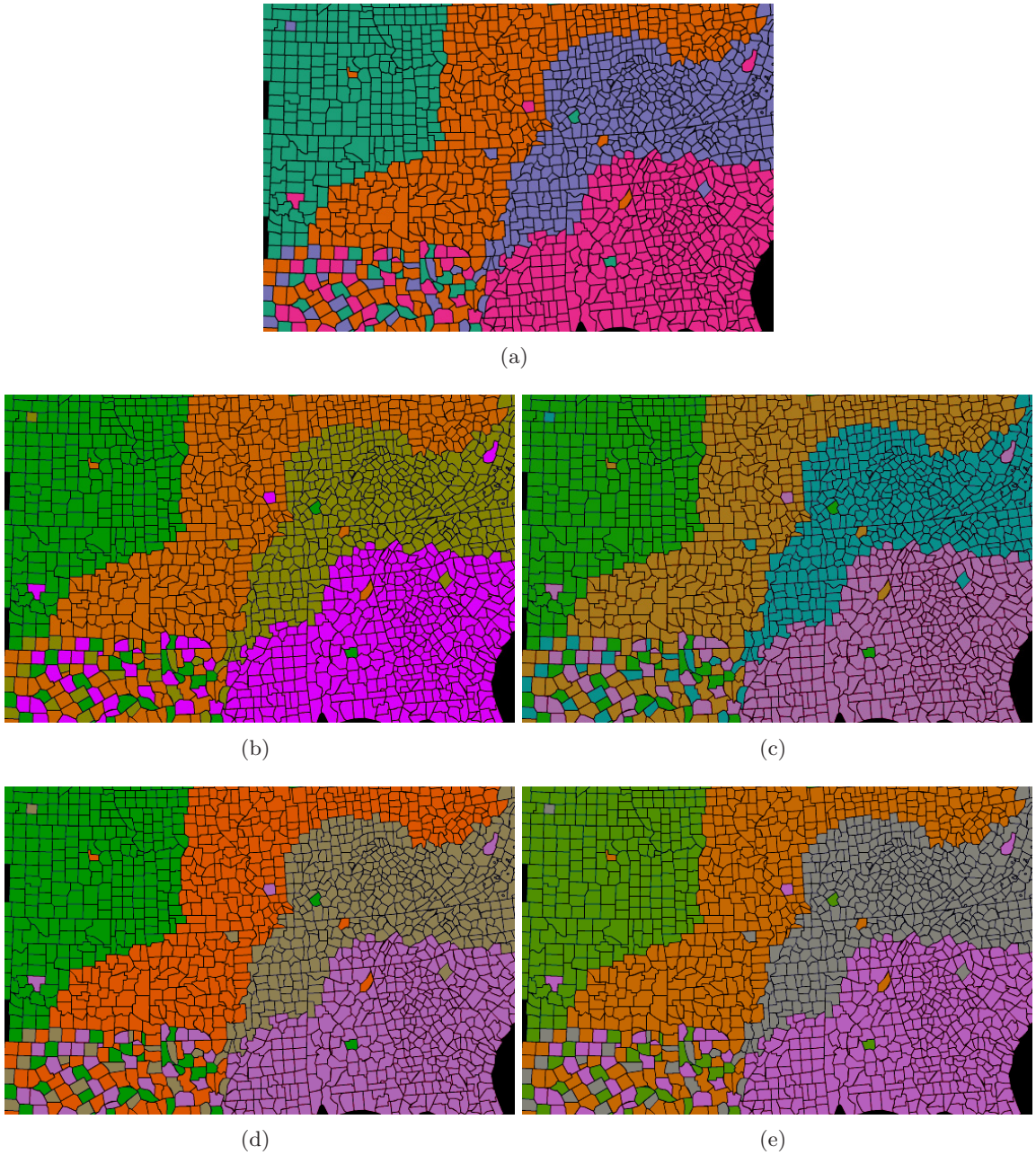


Figure 5.3: (a) Result from the hybrid optimization with constrained saturation (i.e. no gray) using input parameters $N = 4$, $L^* = 53.90$, $d = 63.80$, and $S = 0.45$. (b) Result from the hybrid optimization with unconstrained saturation (i.e. gray allowed) using input parameters $N = 4$, $L^* = 53.90$, $d = 63.80$, and $S = 0$.



(a)

(b) (c)

(d) (e)

Figure 5.4: Visual comparison of the 2D ColorBrewer map using different energy aware colour sets. (a) Colours from the original *4-class qualitative Dark2* scheme provided by ColorBrewer. Energy aware colours from (b) discrete optimization, (c) continuous optimization, (d) hybrid optimization without gray, and (e) hybrid optimization with gray allowed. (a)—(c)

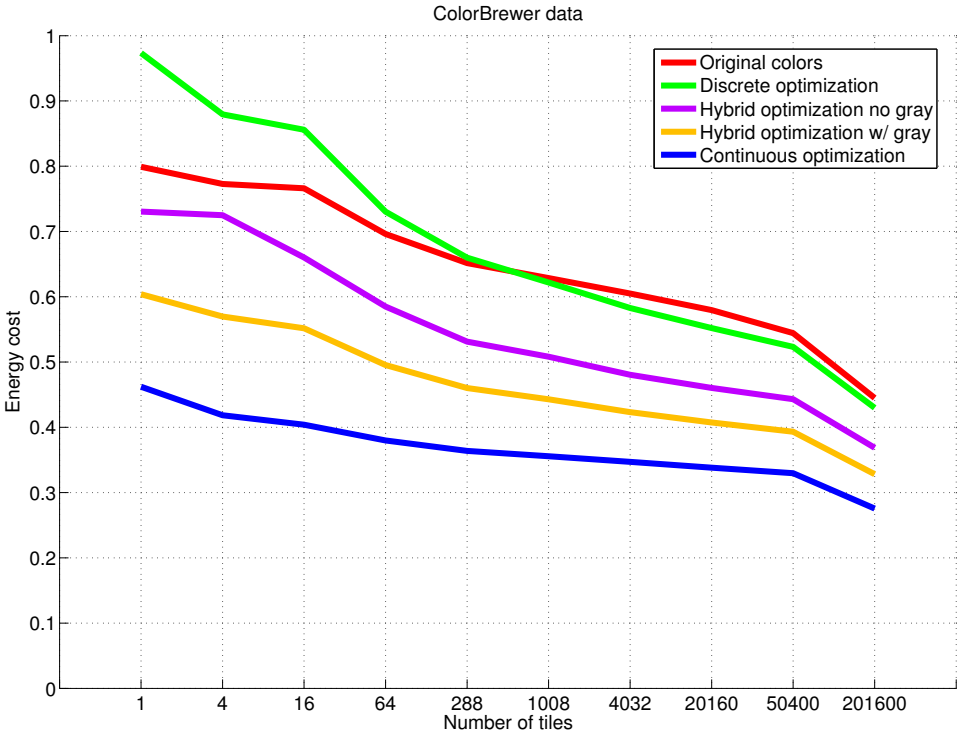


Figure 5.5: Energy comparison of the 2D ColorBrewer map using different energy aware colour sets. On a display using 760 tiles [36], up to 44 percent of the energy is saved by using colours from the continuous optimization.

5.2 Energy Aware Colours for Volume Rendering

The second visualization application is from volume visualization. Figure 5.6 shows an example of the volume rendering of a tooth dataset. The images are generated using raycasting with post-classification via 1D transfer functions [10]. The tooth dataset has 3 distinct regions: the crown, outer encasing, and interior. The 3 regions are classified by assigning 3 different colours to the respective scalar value ranges within the transfer function. We want to choose a fair, traditional set of 3 base colours to compare our energy aware colours against. This proves to be difficult, as there is no rule-of-thumb for picking traditional base colours. We decide to use shades of red, green, and blue each at 75% RGB intensity for comparison because colour pickers in typical volume-visualization software are based on RGB and users tend to apply pure RGB colours. A summary of the traditional colour choice is listed in Table 5.6. The average lightness of those RGB colours, $L^* = 44.50$, along with the minimum colour distance and minimum saturation will be the input parameters for our optimizations.

In the first approach using discrete optimization, we use the lookup table (see Figure 3.3) to choose yellow, green, and orange at $L^* = 44.50$. A summary of those colours are listed in Table 5.7. Once again, there are 3 factorial ways of assigning colours, so we try to choose an assignment that looks closest to the original. In this case, we assign green to the outer encasing, yellow to the interior, and orange to the crown. The result can be seen in Figure 5.9(b).

In the second approach using continuous optimization over the continuous CIELAB colour space, we optimize using the parameters $N = 3$ and $L^* = 44.50$. The smallest colour distance between the original red, green and blue colours is 137.02. This minimum distance pushes colours to the boundaries of the iso-lightness plane, forcing undesired high energy colours. Since our goal with the parameter d is to maintain colour distinguishability, we estimate a better, more appropriate d by dividing the length of the gamut boundary of the $L^* = 44.50$ iso-lightness slice (see Figure 3.4(a)) by 7, the maximum number of recommended isoluminant colours [16]. This gives us a minimum distance $d = 70$ for the continuous optimization. Even though this new d is still quite large, it does not push colours to the gamut boundaries. We can see the result of the optimization in Figure 5.7 and a summary of those colours in Table 5.8. The colours are applied to the tooth to produce Figure 5.9(c).

In the third approach using hybrid optimization, we have 2 different sets of 3 colours, one without gray (i.e. constrained saturation) and one with gray (i.e. unconstrained saturation). For the optimization without gray, the input parameters are $N = 3$, $L^* = 44.50$, $d = 70.00$, and $S = 1.00$. $S = 1.00$ is chosen to match the minimum saturation of the traditional colours. In this case, all 3 colours are fully saturated. For the optimization with gray, the input parameters are $N = 3$, $L^* = 44.50$, $d = 70.00$, and $S = 0$. The results from both optimizations are shown in Figure 5.8 and a summary of the colours is listed in Tables 5.9 and 5.10. The colours are applied to the tooth model to produce Figures 5.9(d) and 5.9(e).

Figure 5.9 compares the volume rendered tooth using the traditional choice of colours and energy aware colours. From visual inspection, colours appear to be similarly discernible in Figure 5.9(a) (i.e. the original RGB choice) and Figure 5.9(c) (i.e. continuous optimization). However, in Figures 5.9(b) (i.e. discrete optimization), 5.9(d) and 5.9(e) (i.e. hybrid optimization), it is hard to distinguish the inner structure of the tooth. This is due to the effects of traditional colour blending, causing colours to mix unpredictably and consequently decrease colour distinguishability. By using the hue-preserving blending method as described in Chapter 4, colour distinguishability for the differently classified regions is improved, as shown in Figure 5.11.

The tooth data set is automatically rotated to simulate typical volume interaction, and 11 separate energy measurements are taken at regular intervals, then averaged. Figure 5.10 documents the average energy consumption from the volume interaction, showing an energy saving of 45 percent using colours from the continuous optimization on a practical display using 760 tiles. We repeat the experiments with hue-preserving blending and document the energy consumption in Figure 5.12. Hue-preserving blended renderings tend to use even lower energy (but not always; compare the energy of Figure 5.9(a) and Figure 5.11(a)), in addition to improving colour distinguishability. In general, we can expect hue-preserving blending to use no more energy than non-hue-preserving blending because hue-preserving blending aims to preserve the original lightness, and energy monotonically increases with respect to lightness. We can also expect hue-preserving blending to use less energy, because colours tend to become desaturated (as colours blend toward gray), and desaturated gray colours have the least energy in an iso-lightness slice. The energy comparison of Figure 5.9(a) and Figure 5.11(a), however, is different than expected (the hue-preserving uses slightly higher energy than non-hue-preserving). This is because adding lightness components in the HSL space (i.e., $L_1 + L_2$, as we do in our new blending model), may not be the most accurate

lightness computation for blending two colours. Consequently this may lead to lightness errors and slightly higher energy consumption. A more perceptually accurate colour model may be used for more accurate lightness computation. Nevertheless, HSL has proven to be easy to understand and use in practice, and we still observe energy improvements in the other tooth comparisons.

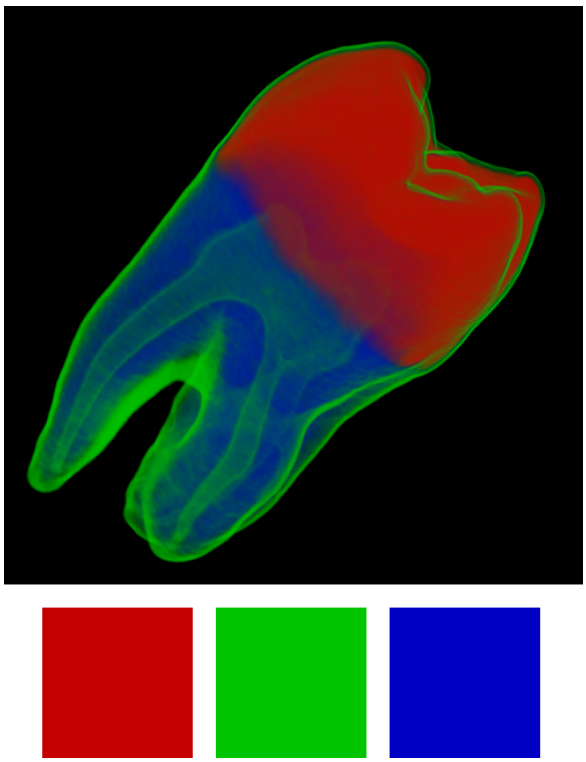


Figure 5.6: Volume rendering of a tooth dataset using a traditional choice of colours. ©Eurographics Association 2009. Reproduced with kind permission of the Eurographics Association.

	green	blue	red
R	0	0	196
G	196	0	0
B	0	196	0
L^*	69.09	23.61	40.79
a^*	-70.68	65.05	65.72
b^*	68.23	-88.51	55.14
saturation	1.00	1.00	1.00
Average L^*: 44.50 Minimum $L^*a^*b^*$ colour distance: 137.02 Maximum $L^*a^*b^*$ colour distance: 207.34			

Table 5.6: Colour summary of the traditional choice of colours for the tooth data set.

	green	yellow	orange
R	0	110	167
G	125	110	84
B	0	0	0
L^*	45.17	44.89	45.02
a^*	-50.81	-11.59	29.89
b^*	49.05	50.85	54.23
saturation	1.00	1.00	1.00
Average L^*: 45.03			
Minimum $L^*a^*b^*$ colour distance: 39.26			
Maximum $L^*a^*b^*$ colour distance: 80.87			

Table 5.7: Colour summary from the discrete optimization for the tooth data set.

	teal	purple	brown
R	0	145	145
G	120	84	96
B	107	144	0
L^*	44.93	44.76	44.78
a^*	-31.65	34.91	13.29
b^*	-0.96	-22.60	52.55
saturation	1.00	0.44	1.00
Average L^*: 44.82			
Minimum $L^*a^*b^*$ colour distance: 69.87			
Maximum $L^*a^*b^*$ colour distance: 78.19			

Table 5.8: Colour summary from the continuous optimization for the tooth data set.

	green	yellow	orange
R	0	117	191
G	122	108	61
B	68	0	0
L^*	44.71	44.95	44.82
a^*	-42.52	-7.10	50.12
b^*	21.56	51.19	56.27
saturation	1.00	1.00	1.00
Average L^*: 44.83			
Minimum $L^*a^*b^*$ colour distance: 46.18			
Maximum $L^*a^*b^*$ colour distance: 98.93			

Table 5.9: Colour summary from the hybrid optimization (with constrained saturation) for the tooth data set.

	green	gray	orange
R	0	106	183
G	124	106	70
B	0	105	2
L^*	44.82	44.79	44.87
a^*	-50.52	-0.18	43.35
b^*	48.77	0.55	54.95
saturation	1.00	0.011	0.99
Average L^*: 44.83			
Minimum $L^*a^*b^*$ colour distance: 69.68			
Maximum $L^*a^*b^*$ colour distance: 94.07			

Table 5.10: Colour summary from the hybrid optimization (with unconstrained saturation) for the tooth data set.

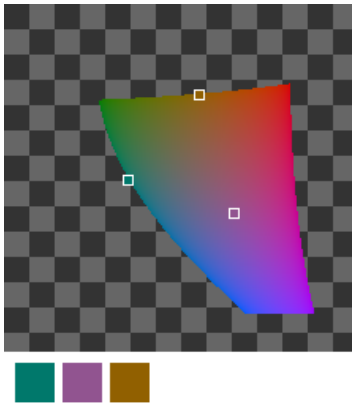


Figure 5.7: Result from the continuous optimization using input parameters $N = 3$, $L^* = 44.50$, and $d = 70.00$.

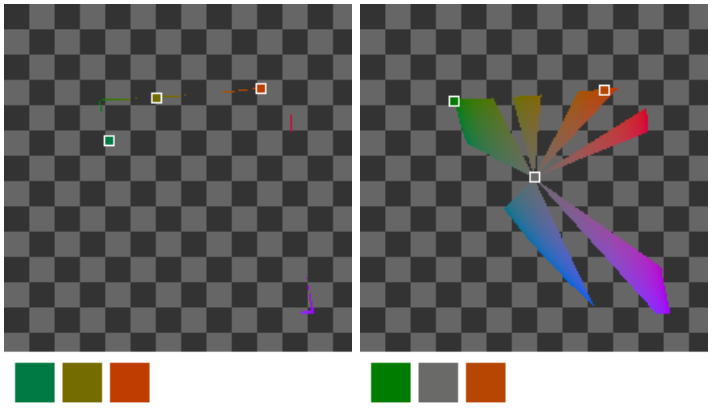


Figure 5.8: (a) Result from the hybrid optimization with constrained saturation (i.e. no gray) using input parameters $N = 3$, $L^* = 44.50$, $d = 70.00$, and $S = 1.00$. (b) Result from the hybrid optimization with unconstrained saturation (i.e. gray allowed) using input parameters $N = 3$, $L^* = 44.50$, $d = 70.00$, and $S = 0$.

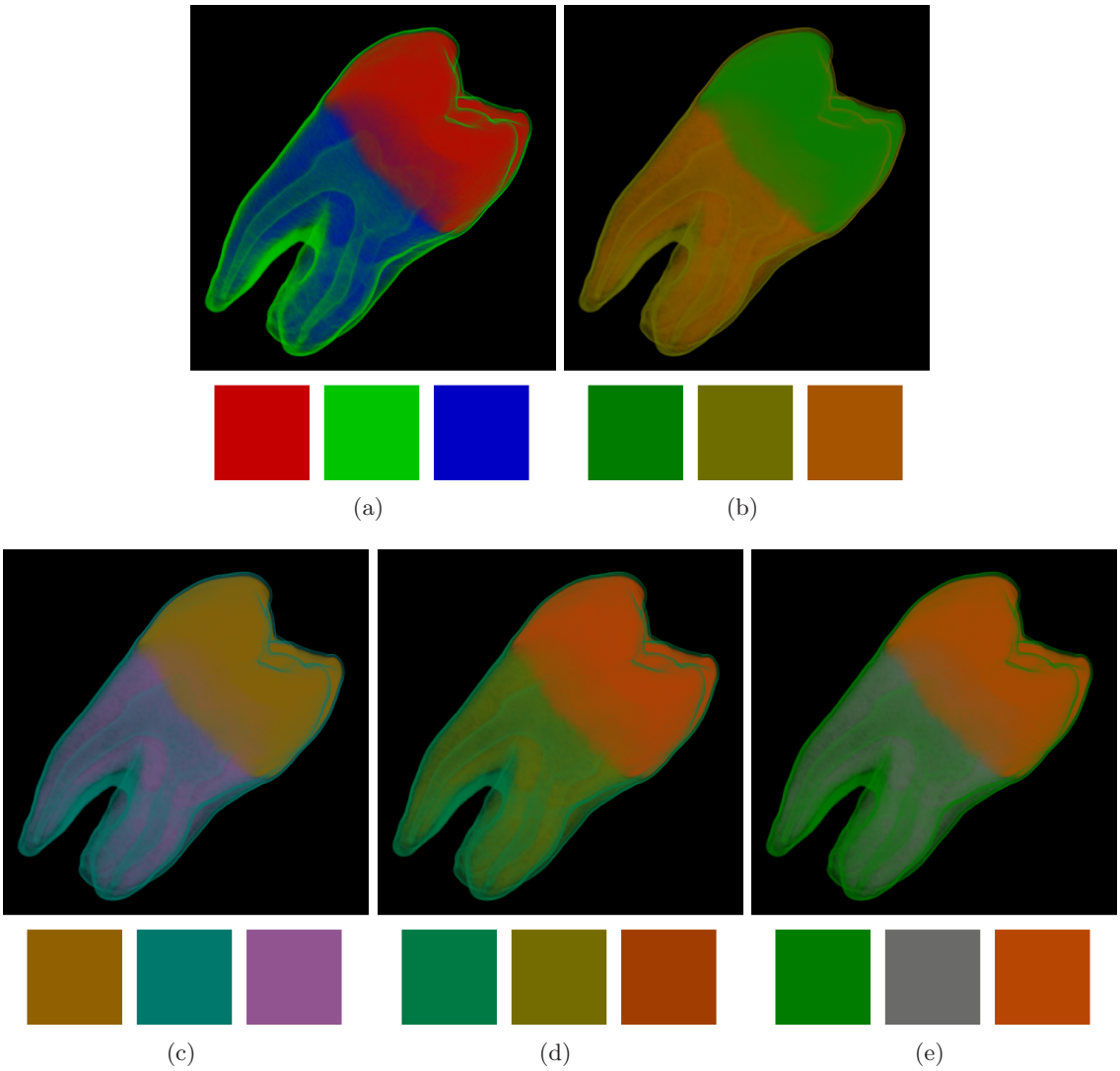


Figure 5.9: Visual comparison of the tooth dataset using (a) traditional colours, energy aware colours from (b) discrete optimization, (c) continuous optimization, (d) hybrid optimization without gray, and (e) hybrid optimization with gray allowed. The coloured squares represent the discrete colours used for volume classification. (a)—(c)

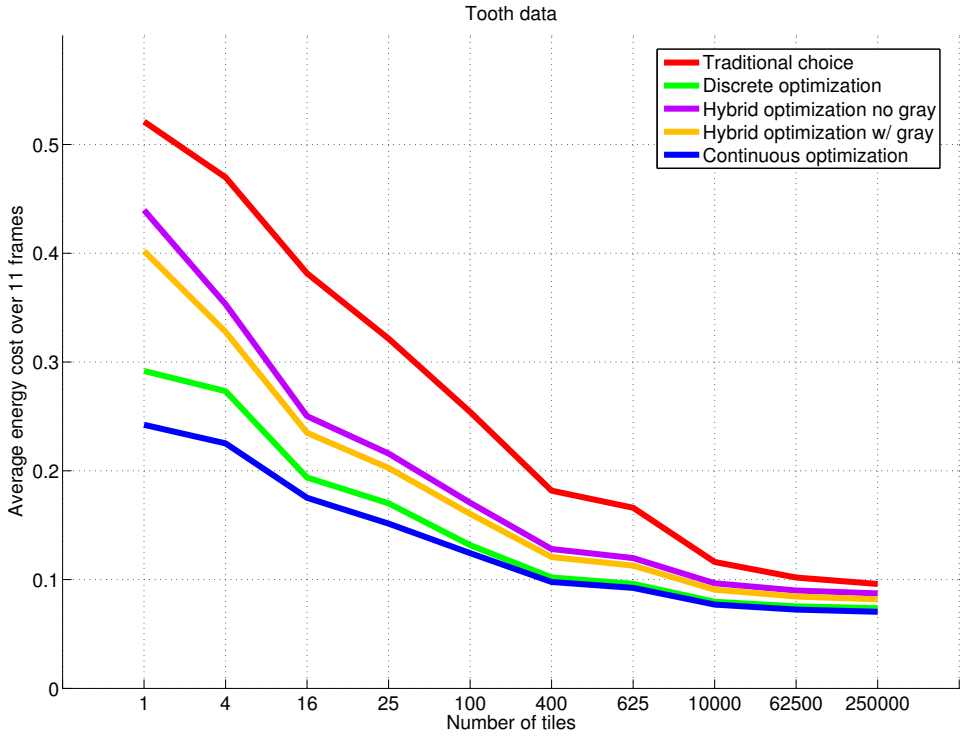


Figure 5.10: Energy comparison of the volume rendered tooth using different colour sets. On a display using 760 tiles [36], up to 45 percent of the energy is saved by using colours from the continuous optimization.

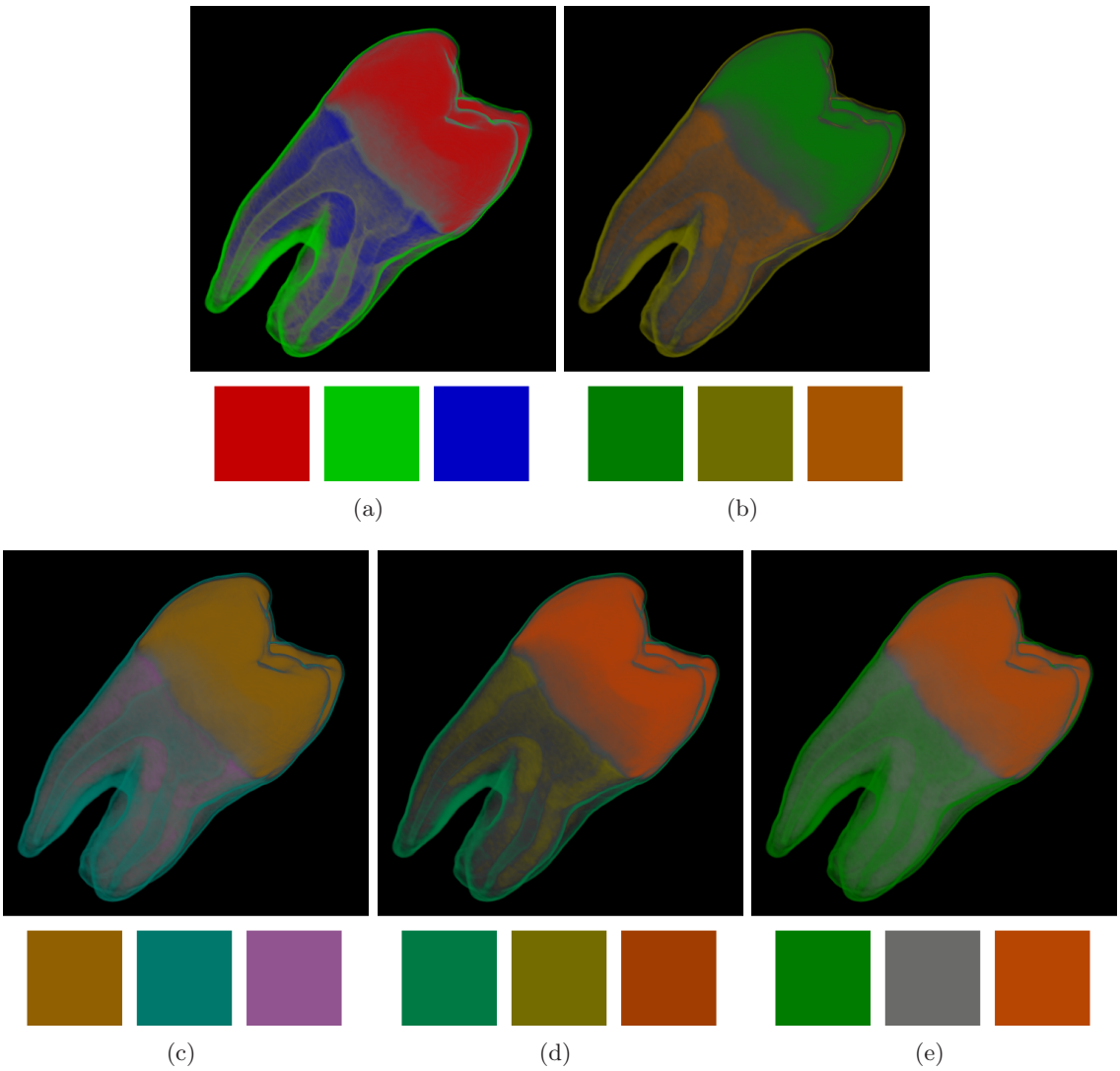


Figure 5.11: Visual comparison of the hue-preserving volume rendered tooth dataset using (a) traditional colours, energy aware colours from (b) discrete optimization, (c) continuous optimization, (d) hybrid optimization without gray, and (e) hybrid optimization with gray allowed. The coloured squares represent the discrete colours used for volume classification. Compare these images with those of Figure 5.9.

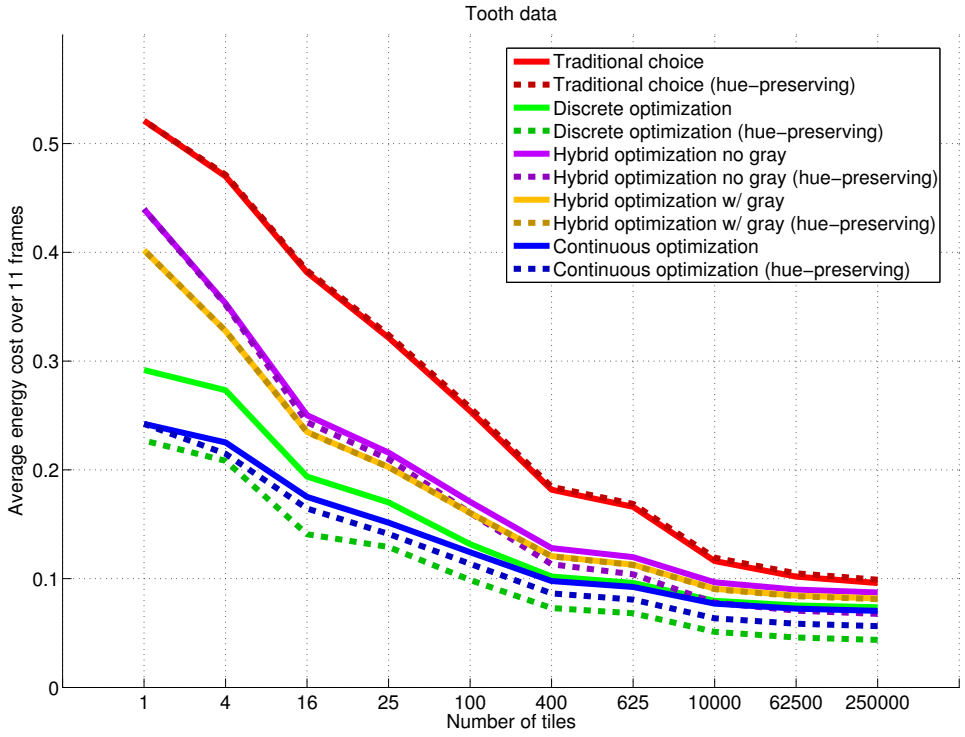


Figure 5.12: Energy comparison of the hue-preserving volume rendered tooth using different colour sets. Energy consumption tend to be lower with hue-preserving colour blending.

Chapter 6

Conclusion and Future Direction

The main goal of this thesis is to trigger awareness for the energy consumption associated with emerging colour displays. We have shown that substantial energy savings can be achieved by changing the colour mapping in data visualization—without sacrificing good visual perception. On a technical level, we have presented three design methods for distinguishable, iso-lightness colours with the goal of lowering the energy consumption of the display device. Our methods are based on a screen space variant energy model, using a grid of tiles to estimate energy consumption. The first approach is an optimization of discrete colours based on colour category. We restrict ourselves to using a small set of colour hues, as previous work suggest that having too many colours slows down the speed of colour identification. The second approach optimizes over a continuous iso-lightness colour space by minimizing a cost function that takes into account the energy of the colours and the distances between them. The third method is a hybrid of the first two, optimizing over a set of distinct, continuous, categorical colour ranges. All three methods address colour distinguishability and iso-lightness. We compare our colour mappings to those chosen by the ColorBrewer system for 2D maps and to a traditional set of colours for volume rendering. Typical energy savings are around 45 percent for these examples.

We have also presented hue-preserving blending as a modification of general Porter and Duff image compositing. Hue-preserving colour blending improves visual labeling by colour, even in transparent rendering. Our model is based on results from previous perception research indicating that perceptual transparency may be treated separately for achromatic and chromatic information. Accordingly, we have reused existing blending models for the achromatic channel and just modified chromatic compositing. Here, the main idea is to

identify the dominating colour whose hue survives blending; continuous colour transition is achieved by gradually changing saturation, instead of hue. Several examples of volume-rendered images using traditional and hue-preserving blending are compared. In some cases, hue-preserving blending can lead to even lower display energy consumption.

There are several challenges in designing energy aware colours. The discrete optimization approach might suffer from the problem that colours are not chosen according to perceptual distance. In particular, modifying luminance may lead to similar looking colours. For example, dark shades of orange and yellow look very similar because brown is considered a dark shade of yellow or orange alike. Therefore, additional control over perceptual distance would be beneficial for the discrete optimization approach. In contrast, the continuous optimization of colours provides explicit control over perceptual colour distance and, thus, does not run into the above issue of the discrete approach. However, the continuous optimization is not aware of colour names or categories. Therefore, we can end up having more than one colour from the same category (see Figure 3.6(a) for an example with 3 greenish colours), which might be problematic if visualization is used for visual communication that requires naming of image regions. The third hybrid approach attempts to resolve the problems of the first two, but as seen in energy measurements for the tooth dataset (Figure 5.10), it can result in higher energy costs compared to the continuous optimization (because there are additional constraints), and higher energy costs compared to the discrete optimization (because of the minimum colour distance requirement).

There are other possible future directions of our work. A practical benefit of our hue-preserving blending approach is that it may be readily included in any visualization system using Porter and Duff compositing because only minimal algorithmic changes are required. We have targeted direct volume visualization as the main application, but any kind of non-photorealistic image overlay may benefit, too. Future research might also explore alternative colour (appearance) models. Perceptual effects like simultaneous contrast and surround luminance certainly have not been addressed in our work. Similarly, visual inspection of our results ought to be followed by a full validation study, including actual power measurements and colour naming tests, in order to assess the effectiveness of our colour designs for specific visualization applications.

Our methods can also be used to design energy-friendly GUIs that use energy aware colour schemes and still maintain content readability. Low energy colour GUIs can be beneficial for battery-powered mobile devices, prolonging their battery life. Furthermore,

we are not limited to iso-lightness colours; we only choose to use iso-lightness colours in our work as a design constraint for our colour maps. One can imagine the use of non-iso-lightness colours for the design of colour maps. An optimization can be performed in a CIELAB subspace, bounded by user-specified L^*_{min} and L^*_{max}, to explore colours with varying lightness. Finally, the long-term goal is a general method to automatically convert arbitrary input images to images with energy aware colours.

Bibliography

[1] B. L. Anderson. A theory of illusory lightness and transparency in monocular and binocular images: The role of contour junctions. *Perception*, 26(4):419–452, 1997.

[2] J. Beck, K. Prazdny, and R. Ivry. The perception of transparency with achromatic colors. *Perception & Psychophysics*, 35(5):407–422, 1984.

[3] D. Borland and R.M. Taylor. Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17, 2007.

[4] V. J. Chen and M. D'Zmura. Test of a convergence model for color transparency perception. *Perception*, 27(5):595–608, 1998.

[5] W. Cheng and C. Chao. Perception-guided power minimization for color sequential displays. In *ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 290–295, 2006.

[6] I. Choi, H. Shim, and N. Chang. Low-power color TFT LCD display for hand-held embedded systems. In *International Symposium on Low Power Electronics and Design (ISLPED)*, pages 112–117, 2002.

[7] J. Chuang, D. Weiskopf, and T. Möller. Energy aware color sets. *Computer Graphics Forum (Eurographics 2009)*, 28(2):203–211, 2009.

[8] Dolby. Technology behind Dolby HDR video, 2008. http://www.dolby.com/professional/video/hdr-video-tech-overview.html, [last access: 29 Sep 2008].

[9] M. D'Zmura, P. Colantoni, K. Knoblauch, and B. Laget. Color transparency. *Perception*, 26(4):471–492, 1997.

[10] K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, and D. Weiskopf. *Real-Time Volume Graphics*. A K Peters, 2006.

[11] M. D. Fairchild. *Color Appearance Models*. Wiley & Sons, 2nd edition, 2006.

[12] S. R. Forrest. The road to high efficiency organic light emitting devices. *Organic Electronics*, 4(2–3):45–48, 2003.

[13] W. Gerbino, C. I. Stultiens, J. M. Troost, and C. M. de Weert. Transparent layer constancy. *Journal of Experimental Psychology. Human Perception and Performance*, 16(1):3–20, 1990.

[14] M. A. Harrower and C. A. Brewer. ColorBrewer.org: an online tool for selecting color schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[15] T. Harter, S. Vroegindeweij, E. Geelhoed, M. Manahan, and P. Ranganathan. Energy-aware user interfaces: an evaluation of user acceptance. In *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 199–206, 2004.

[16] C. G. Healey. Choosing effective colours for data visualization. In *Proceedings of the IEEE Conference on Visualization*, pages 263–270, 1996.

[17] A. Iranli and M. Pedram. DTM: dynamic tone mapping for backlight scaling. In *DAC '05: Proceedings of the 42nd Annual Conference on Design Automation*, pages 612–617, New York, NY, 2005. ACM.

[18] S. Iyer, L. Luo, R. Mayo, and P. Ranganathan. Energy-adaptive display system designs for future mobile environments. In *MobiSys '03: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pages 245–258, 2003.

[19] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, editors. *Essentials of Neural Science and Behavior*. Appleton & Lange, Norwalk, 1995.

[20] R. Kasrai and F. A. A. Kingdom. Precision, accuracy, and range of perceived achromatic transparency. *Journal of the Optical Society of America A*, 18(1):1–11, 2001.

[21] M. Kawai, K. Uchikawa, and H. Ujike. Influence of color category on visual search. In *Annual Meeting of the Association for Research in Vision and Ophthalmology (Fort Lauderdale)*, 1995. Paper #2991.

[22] G. Kindlmann, E. Reinhard, and S. Creem. Face-based luminance matching for perceptual colormap generation. In *Proceedings of the IEEE Conference on Visualization*, pages 299–306, 2002.

[23] G. R. Kuhn, M. M. Oliveira, and L. A. F. Fernandes. An efficient naturalness-preserving image-recoloring method for dichromats. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1747–1754, 2008.

[24] Z. Lu, L. Lesmes, and G. Sperling. Perceptual motion standstill from rapidly moving chromatic displays. In *Proceedings of National Academy of Science, 96*, pages 15374–15379, 1999.

[25] F. Metelli. The perception of transparency. *Scientific American*, 230(4):91–98, 1974.

[26] B. Mochocki, K. Lahiri, and S. Cadambi. Power analysis of mobile 3D graphics. In *DATE '06: Proceedings of the Conference on Design, Automation and Test in Europe*, pages 502–507. European Design and Automation Association, 2006.

[27] M. Moser and D. Weiskopf. Interactive volume rendering on mobile devices. In *Vision, Modeling, and Visualization VMV '08 Conference Proceedings*, pages 217–226, 2008.

[28] V. G. Moshnyaga and E. Morikawa. LCD display energy reduction by user monitoring. *ICCD '05: Proceedings of the 2005 International Conference on Computer Design*, pages 94–97, 2005.

[29] S. Murugesan. Harnessing green IT: Principles and practices. *IT Professional*, 10(1):24–33, 2008.

[30] K. Nakayama, S. Shimajo, and V. S. Ramachandran. Transparency: Relation to depth, subjective contours, luminance, and neon color spreading. *Perception*, 19(4):497–513, 1990.

[31] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[32] T. Porter and T. Duff. Compositing digital images. *Computer Graphics (ACM SIGGRAPH 1994 Conference)*, 18(3):253–259, 1984.

[33] V. S. Ramachandran and R. L. Gregory. Does colour provide an input to human motion perception? *Nature*, 275:55–57, 1978.

[34] P. Ranganathan, E. Geelhoed, M. Manahan, and K. Nicholas. Energy-aware user interfaces and energy-adaptive displays. *Computer*, 39(3):31–38, 2006.

[35] P. Rheingans and B. Tebbs. A tool for dynamic explorations of color mappings. In *Symposium on Interactive 3D Graphics*, pages 145–146, 1990.

[36] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trentacoste, A. Ghosh, and A. Vorozcovs. High dynamic range display systems. *ACM Transactions on Graphics*, 23(3):760–768, 2004.

[37] F. Shearer. *Power Management in Mobile Devices*. Newnes, Burlington, MA, 2008.

[38] P. Tsai, C. Liang, and H. H. Chen. Image quality enhancement for low backlight TFT-LCD displays. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 473–476, 2007.

[39] K. S. Vallerio, L. Zhong, and N. K. Jha. Energy-efficient graphical user interface design. *IEEE Transactions on Mobile Computing*, 5(7):846–859, 2006.

[40] M. A. Viredaz, L. S. Brakmo, and W. R. Hamburgen. Energy management on handheld devices. *Queue*, 1(7):44–52, 2003.

[41] L. Wang, J. Giesen, K. T. McDonnell, P. Zolliker, and K. Mueller. Color design for illustrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1739–1754, 2008.

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

[43] C. Ware. *Information Visualization.* Morgan Kaufmann, 2nd edition, 2004.

[44] J. Williams and L. Curtis. Green: The new computing coat of arms? *IT Professional*, 10(1):12–16, 2008.

[45] L. Zhong and N. K. Jha. Graphical user interface energy characterization for handheld computers. In *CASES '03: Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems*, pages 232–242, New York, NY, 2003. ACM.