

Using and Adapting to Limits of Human Perception in Visualization

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Nicholas Waldin

Matrikelnummer 1429048

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Dr. Ivan Viola

Diese Dissertation haben begutachtet:

Karol Myszkowski

Petra Isenberg

Wien, 24. August 2017

Nicholas Waldin

Using and Adapting to Limits of Human Perception in Visualization

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Nicholas Waldin

Registration Number 1429048

to the Faculty of Informatics

at the TU Wien

Advisor: Dr. Ivan Viola

The dissertation has been reviewed by:

Karol Myszkowski

Petra Isenberg

Vienna, 24th August, 2017

Nicholas Waldin

Erklärung zur Verfassung der Arbeit

Nicholas Waldin
LinzerStrasse 429/2011, 1140 Wien, Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 24. August 2017

Nicholas Waldin

Danksagung

Manuela Waldner, wiederkehrende Mitautorin. Meine Kollegen, insbesondere Alexey und Viktor, sowie alle andere Mitarbeiter.

Acknowledgements

Manuela Waldner, who was a recurring co-author. My friends, especially Alexey and Viktor. And all the other people who I worked or discussed with.

Kurzfassung

Im Zuge der Analyse einer Visualisierung müssen Benutzer oft wichtige Objekte finden oder vergleichen. Dabei werden die Benutzer mit einem grundlegenden Problem konfrontiert. Die zunehmend großen Datenmengen führen zu visueller Übersättigung (“visual clutter”). Die Objekte, in die der Benutzer interessiert ist, werden von vielen anderen Objekten umgeben, womit es schwierig ist, die wichtigen Objekte zu finden. Dieses Problem entsteht teilweise durch die Limitierungen des menschlichen visuellen Systems, welches sowohl durch die Bandweite des sichtbaren Lichts, die visuelle Auflösung, als auch durch die Prozess-Kapazität des menschlichen Gehirns beschränkt ist.

In dieser Dissertation werden drei Methoden gezeigt, wie diese Limitierungen erweitert oder auch positiv genutzt werden können. Die erste Methode wurde entwickelt, um Personen mit Farbenblindheit, wie zum Beispiel rot-grün Blindheit, eine personalisierte, erweiterte Farbpalette zu bieten. Personen mit Farbenblindheit haben Schwierigkeit verschiedene Farben zu unterscheiden. Ziel unserer Methode ist, die Farbkarte eines Datensatzes an individuelle Personen anzupassen und damit deren persönlichen Farbraum zu maximieren. Die zweite Methode bietet eine dynamische Erweiterung des Farbraums für große hierarchische Daten. Während der interaktiven Exploration der Szene wird der Farbraum dynamisch an den Blickpunkt angepasst. Dabei hilft die “Unaufmerksamkeitsblindheit” – d.h., das Ignorieren von Veränderungen in der Szene, die nicht im Fokus des Betrachters liegen – um die Farbänderungen größtenteils unbemerkt durchführen zu können. Die dritte Methode benutzt Flimmern um Teile einer Szene möglichst subtil hervorzuheben. Dabei nutzen wir die Tatsache, dass die “Flimmerverschmelzungsfrequenz” – d.h. die Frequenz, ab welcher Flimmern als stabiles Signal wahrgenommen wird – über die Retina variiert. Durch die Benutzung eines hochfrequenten Monitors und empirischer Messungen von Flimmerparametern konnten wir eine Methode zur Hervorhebung von Objekten finden, die nur in der visuellen Peripherie sichtbar ist, nicht aber in der Fovea.

Abstract

When analyzing a visualization, the user must often find or compare important objects. This analysis suffers from a fundamental problem: data sets are becoming larger and larger, leading to more visual clutter. This makes it very hard to find the objects the user is interested in. Part of this problem originates in the human visual system, which is limited through the bandwidth of visual light, visual resolution, and the processing capabilities of the human mind.

In this thesis, three methods are shown that adapt to these limitations, and use them to the advantage. The first method targets people with color vision deficiency (CVD), such as red-green blindness. People with CVD have difficulty discerning colors. The aim of this method is to adapt a color map to the individual and maximize the use of their personal color space. The second method offers a dynamic use of the color space for large hierarchical data. During interactive exploration of the data, the color mapping adapts on-the-fly to the current view position. We make use of “inattention blindness”—i.e., not noticing changes that are not focused on—in order to make the change in color very subtle. The third method uses flicker in order to subtly draw attention to parts of a scene. We use the fact that the “critical fusion frequency”—the frequency at which flickering becomes a stable signal—varies across the retina. Using a high frequency monitor and empirical measurements, we created a method that can draw attention to objects and can only be seen in the peripheral vision, but not in the foveal vision.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
1 Introduction	1
1.1 Scope	3
1.2 Contributions and Co-authors	3
1.3 Thesis Structure	4
2 Related Work	5
2.1 Human Vision	5
2.2 Color Vision Deficiency Types and Effects	8
2.3 Genetics and Sex Differences	9
2.4 Color Maps	9
2.5 Recoloring for CVD	12
2.6 Biomolecular Visualization	15
2.7 Flicker Physiology	15
2.8 Flicker Application	17
3 Verbose Contribution	19
3.1 Paper A: Personalized 2D Color Maps	21
3.2 Paper B: Chameleon	23
3.3 Paper C: Cuttlefish	25
3.4 Paper D: Flicker Observer Effect	27
4 Conclusion and Future Work	29
A Personalized 2D Color Maps	31
B Chameleon: Dynamic Color Mapping for Multi-Scale Structural Biology Models	41
	xv

C Cuttlefish: Color Mapping for Dynamic Multi-Scale Visualizations	53
D Flicker Observer Effect: Guiding Attention Through High Frequency Flicker in Images	67
List of Figures	79
Bibliography	81

Introduction

The human mind is limited in its capabilities to understand data, both in scope and in form. A simple example of the latter is a series of numbers, of which a person must find the three largest. This is a difficult task if the user is required to use a textual form, but simple with a bar-chart visualization. Unfortunately, the human visual system is extremely complicated and not yet completely understood. Some of the rules it plays by are not obvious. The dual search task shown in Figure 1.1 is one such case. A person can quickly detect an item of different color or shape from among a large number of similar items, such as a red circle among green circles, or a square among circles. In fact, the time required is barely connected to the number of items. However, if a person is asked to find a green square among a set of green circles and red squares, then the amount of time is linearly dependent on the number of items. This vast difference in speed indicates the necessity of understanding and adapting to the rules visual perception adheres to.

The limitations are not restricted to such basic phenomena. Problems can also be found at a higher level, such as concentration. A well-known phenomenon is inattentive blindness—often also called “tunnel-vision”—where a person will pay attention to a specific event or item, while unwillingly ignoring other important items, possibly even when he would like to notice them. An example of inattentive blindness can be demonstrated by showing a video of people passing a basketball back and forth and asking a person to count the number of passes. During this video, a

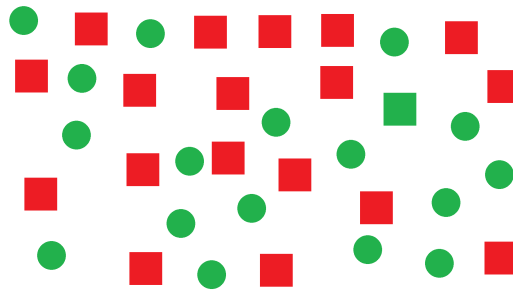


Figure 1.1: Example of the dual search task. There is one green square, surrounded by red squares and green circles.

person in a gorilla outfit can walk across the scene without being noticed [SC99]. There is also the case of change blindness, where a person will not detect a difference when a scene is shown and then swapped out for a version with a slight change. In fact, this has even become a form of game, often found in newspapers and similar, where the reader is asked to spot the differences in two images. These problems have different causes, though they are connected. In order to explain the first case, a quick explanation of the visual system is necessary. The visual system can be viewed as different layers. At the bottom are very simple mechanisms, such as edge detectors. These pass a signal further up to the next layer, and so forth. However, signals can also be sent back down, and signals are “louder” or “quieter” depending on what the higher layers are looking for. In the case of the gorilla costume, the signals of the gorilla costume are too quiet in comparison to the basketball. Change blindness however is affected by the working memory. A person can only keep 4-8 items in his “working memory”, i.e., remember what they look like. Therefore, if there is a change outside of this limited number, he will not be able to detect it.

Given all of these problems, it is no surprise that the visual system can be seen as limiting, rather than useful, though it is common to see methods that take advantage of the strengths of the visual system in order to draw a person's attention to a specific point. The visual system is good at finding areas that are significantly lighter or darker for example, so highlighting a point on the monitor by making it lighter than the surrounding area is a simple way of drawing a user's attention. Other approaches attempt to use the ability of finding the “odd one out” by using stick figures to encode data [PG88]. This approach uses the same principle as finding a circle among squares. The different orientations are easily differentiable. This also works for large data sets, as it is easy to find patterns or similar items in a specific area, as all the items in that area will have a similar shape. Other approaches use the human ability to detect and understand faces easily. Similar to the stick figure approach, data is encoded as different types of faces [Che73], where each dimension will be encoded as an aspect of the face. For example, a large scalar value can be encoded as “angry eyes”, while a low value becomes “happy eyes”. Similar to the stick figure approach, patterns and similarities are easy to spot. A completely different application of face recognition was used by [KRC02a], where a person was asked to compare two faces and alter them until they were equi-luminant. This allowed for the creation of an equiluminant color bar. Methods for using the principles of perception are of course used in other approaches as well. For example, in stream surface visualization Carnecky et al. [CFM⁺13] used the way edge detection is done by the visual system to improve the perception of convoluted stream surfaces. To show this he measured how well a person could estimate on which layer a point was, or how far away two different points were on the surface.

However, the research in this thesis takes an alternative approach. The previous methods identified an aspect of the visual system that helps identify items and used it to increase the speed or quality of data recognition. Rather than play to the strengths of the visual system, our research is aimed at the weaknesses. A good example is what we call the

“Flicker Observer Effect”. The sensitivity to flicker varies across the retina and forms an inverse ‘U’ from fovea to periphery. This means that it is possible to detect flicker in the peripheral, but not in the foveal vision, given the right parameters. This difference in flicker, especially the reduced detection in the foveal vision, can be considered a form of weakness. However, it can be used to direct the attention of the users to a specific point, without them being able to see the flicker when they look directly at it, creating a new form of highlighting. It is in this manner that we have tried to start a new way of integrating the visual system into the creation of a visualization.

1.1 Scope

Three methods in the previously described manner are contained in this thesis. The first is aimed at people with Color Vision Deficiency (CVD), described in Paper A. Here a novel method for adapting color maps to a person with CVD is shown, as well as a novel method for measuring the perceptual difficulties a person has. This method can be applied to any monitor and adapts the coloring to the individual. The second is a method for adaptive coloring with high data density that is structured in a hierarchical manner, described in Paper B and Paper C. The third (Paper D) is a method for guiding attention in a cluttered image using flicker, where the flicker is not perceivable in the foveal vision. Rather than concentrating on a specific point in the human visual system and finding applications suited to this, we have opted to take visualization problems and find part of the visual system that can be used.

1.2 Contributions and Co-authors

All papers included in this thesis were written during the Ph.D. project, and the main author is also the author of this thesis. Ivan Viola, as the supervisor, co-authored all papers and was primarily involved in the high-level conceptual development. The papers had different co-authors. Paper A was co-authored by Peter Rautek from the King Abdullah University of Science and Technology, as well as Matthias Bernhard, employed at that time by the TU Wien. Peter Rautek was involved in the creation of the idea, and Matthias Bernhard assisted with the development of the user studies. It was published in the Journal “Computers & Graphics” in a Special Section on SCCG 2016. Paper B was co-authored by Eduard Gröller (TU Wien), Mathieu Le Muzic (TU Wien), Manuela Waldner (TU Wien), and David Goodsell and Autin Ludovic from the Scripps Research Institute. The members of the TU Wien were involved in the design and implementation, while the members of the Scripps Research Institute were involved because of their knowledge of biomolecular structures and HIV in particular. The paper was published at the Eurographics Workshop on Visual Computing for Biology and Medicine in 2016. Paper C was co-authored by Eduard Gröller (TU Wien), Mathieu Le Muzic (TU Wien), Manuela Waldner (TU Wien), and David Goodsell, Autin Ludovic and Arthur Olson from the Scripps Research Institute. It is an invited extension of Paper B, and as such the roles of the co-authors was similar. It is currently in submission. Paper D was

co-authored by Manuela Waldner, who was involved in the design of the user studies. It was published at the Eurographics Conference in 2017, and is in the “Computer Graphics Forum” journal.

1.3 Thesis Structure

This Thesis is split into two parts. In the first part, the overall description of the ideas and contribution is given, both for the entire thesis and the individual papers. In the second part, the papers that were published during the studies are listed. Chapter 2 contains the related work and state of the art, as well as information on the human visual system required to understand the papers and approach. Chapter 3 contains a description of the papers, their individual contribution and how they fit into the research direction. Chapter 4 contains the conclusion and possible directions for future work, both in short and long term. Afterwards the published papers are given.

Related Work

This chapter is divided into several parts. The first part is about human vision and information on color vision deficiency (CVD). The second part concerns color maps, as well as adapting images and visualizations to CVD through image recoloring and color map alteration. The third is a small section on biomolecular visualization, followed by a section on flicker.

2.1 Human Vision

Human vision is dependent on two parts: the different cells in the retina and the processing in the brain. The research in this thesis attempts to use and adapt to human perception in various ways. Paper A adapts to the CVD of the individual, and Paper B uses the manner in which a person focuses on information to its advantage. Below are information on these two parts of the human visual system.

2.1.1 Human Retina

When light falls on the retina it is caught by three different types of cone and one type of rod cells. The rod cells are effectively only active during low light situations, meaning that the cone cells are responsible for vision during the day. The cones are referred to as the B (blue), G (green) and R (red) or S (short), M (medium) and L (long) cones, depending on which color or wavelength they are mostly sensitive to. While the distribution and ratios of the cones can vary significantly between individuals, this does not lead to a large effect [HCN⁺05]. It is important to note that there is a large amount of overlap between the cones, with the G and R cone being capable of absorbing light in most of the B-cone spectrum. A cell absorbs light when a photon hits a molecule in the cone and changes it into a different one, a process called bleaching. This discharges an electrical impulse. The impulses travel through the outer plexiform layer to the bipolar, horizontal

2. RELATED WORK

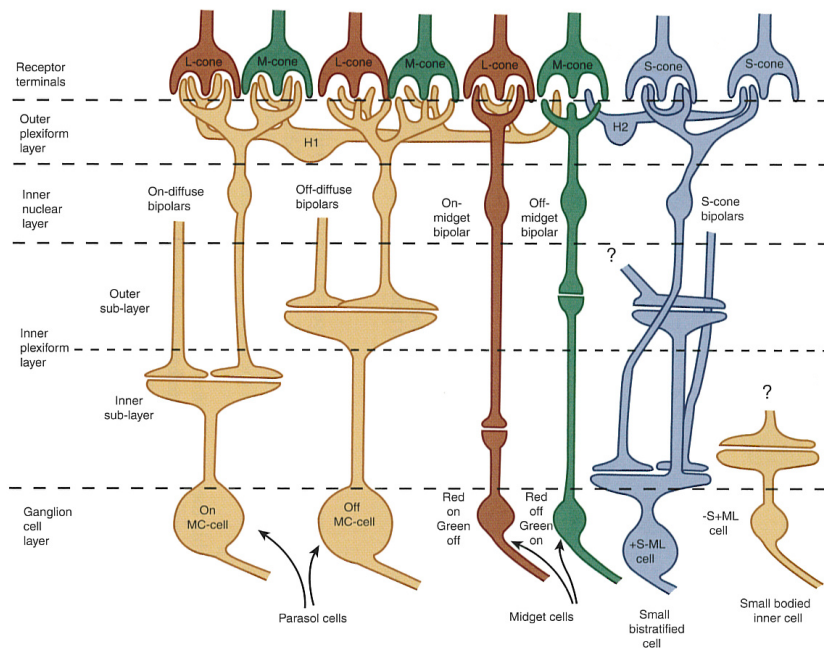


Figure 2.1: Figure 12-12 from [Sch09] by Schwarz. This figure illustrates how the retina of the eye is constructed. The different types of cells and how they are connected are shown. These connections and functions affect how light is interpreted, and need to be taken into consideration when creating a visualization.

and amacrine cells in the inner nuclear layer of the retina. Bipolar cells are diffuse or midget cells. They add up several cones, and the distribution of the light falling on those cones is taken into account (off- and on-center). Midget cells are connected to less cones. The horizontal cells are excited by certain kinds of light and inhibited by others, which affects the color component theory. Amacrine cells tend to respond to stimulus onset and offset. These cell types then send their signal through the inner plexiform layer to the ganglion cells, which reflect the collective properties of the earlier cells. It is important to note that the workings of the retina are quite complex and a detailed description is outside the scope of this thesis. A diagram of this can be seen in Figure 2.1. More information can be found in the book by Schwarz [Sch09].

Horizontal cells can be classified into two classes: H1 and H2 [DR90, Dac96, Mar98]. G and R cones provide most of the input for H1 cells, with only a small amount from B-cones, while H2 cells have a strong connection to B-cones. This creates the basis for color opponency. Color opponency theory has been used for creating perceptual color spaces. A well-known one is CIE Lab, created in 1973. However, it was not completely accurate, and improvements were made by Ebner et al. [EF98], and the international commission on Illumination specified CIECAM02 in 2002. The manner in which humans perceive color has several consequences. For example, it is possible to perceive the same

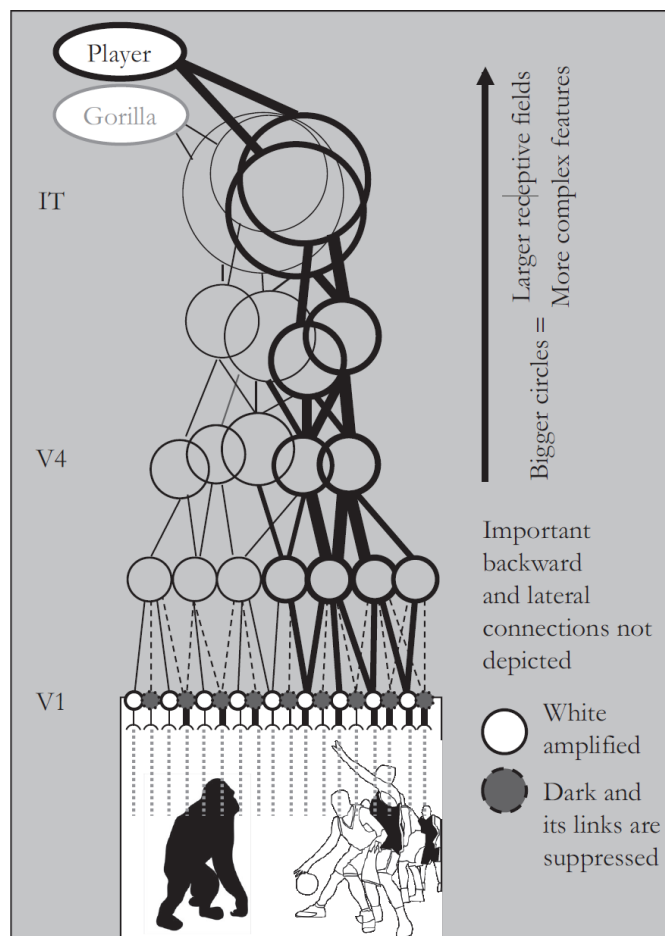


Figure 2.2: Figure 10.2 from *The Nature and Status of Visual Resources* [Fra13] by Franconeri. This image shows the competition for attention in the visual system. Here, the thicker black lines are connected to the players. Because they are the focus of attention, the visual stream regarding the gorilla is not paid any attention, a phenomenon called “inattention blindness”. We use this blindness to our advantage in Paper B.

color even though it is created from different wavelengths. Two colors that are the same but have a different composition are referred to as metamers. Furthermore, increasing the intensity of a light can alter the hue, denoted as the Bezold-Brücke phenomenon. While these phenomena are not necessarily that important when dealing with a computer monitor and a person with normal color vision, they may need to be taken into account when dealing with a person with CVD.

2.1.2 Visual Stream

In the brain, there appear to be two different visual streams, the dorsal and the ventral stream. As the names suggest, the dorsal stream is along the “back” of the brain, going

upwards, while the ventral stream is below it, the “belly” of the brain. The dorsal stream is responsible for determining “where” something is, while the ventral stream determines “what” it is. Both start in the V1 area (visual area 1) of the brain in the striate cortex of the occipital lobe of the cortex. The dorsal stream travels through V2 and V3 to V5, which is also called the middle temporal cortex. The ventral stream travels through V2 to V4, but not V3. This segregation continues further, but becomes more connected and complicated, with the cortex containing at least 20 distinct visual areas [Sch09], and we still do not fully understand the processing of visual information in the brain.

Even though it is called a stream, signals do not flow in just one direction. If a person concentrates on something, then the cells in the visual stream change their response, though this does not appear to be the case for areas prior to V4 [MD85]. This also explains phenomena like change and inattention blindness, such as the gorilla costume experiment [SC99]. In this experiment, people are asked to count how many times a basketball is passed back and forth. Because their attention is focused on the ball, they fail to notice a person in a gorilla costume walk across the screen. Other experiments show that it is possible to switch out the person a test subject is talking to without the subject noticing the swap [SL98]. An illustration of concentration effect is shown in Figure 2.2.

2.2 Color Vision Deficiency Types and Effects

Color Vision Deficiency (CVD) affects roughly eight percent of the male population, and less than 1 percent of the female population. This difference is due to the responsible genes being tied to the X-chromosome. Because of this prevalence, part of the research in this thesis concerned improving visualizations for people with CVD, as it is a weakness in the visual system.

While there are several different types of color deficiencies, they can be placed in two categories: Malfunctioning cone receptors or missing cone receptors. The former is commonly referred to as anomalous trichromacy and, unlike with missing cone receptors, there is a continuous range of severity. The forms of anomalous trichromacy are **Protanomaly**, **Deuteranomaly** and **Tritanomaly**, based on the Greek words for first (prot), second (deuter) and third (trit) and refer to malfunction red cone (R-cone), green cone (G-cone), and blue cone (B-cone) receptors. Likewise, in the case of one missing cone receptor (dichromacy), the cases are called **Protanopia**, **Deuteranopia**, and **Tritanopia**. If two types of cones are missing, then the vision is monochromatic. It is also possible, but extremely rare, that all three cones are missing. When a person suffers from color vision deficiency, then there are equiluminant colors that the person can not distinguish. These colors lie along what are called **confusion lines** that depend on the type of color vision deficiency [Fry92, MB79].

2.3 Genetics and Sex Differences

The three cone cells are controlled by different genes. If a person is missing a cone, then the corresponding gene is either damaged or missing. The reason why a higher portion of the population suffers from red-green-blindness or reduced capabilities is twofold. First, there are two manners in which this occur, with either the R-cone or G-cone missing. Second, the genes for the R and G cone are next to each other and, due to their similarity, have a high chance of genetic mutation during cell splitting [NN11]. Furthermore, this has an effect on anomalous trichromacy. It is still possible for a cone to exist after a mutation occurs, in this case because of a crossover mutation between the two genes. However, the amino acids of the cones have changed. Because of the similarities between the genes, the degree of alteration may be different, which explains how anomalous trichromacy exists on a spectrum. Furthermore, anomalous trichromacy appears to be more complex than a simple reduction of sensitivity. Some studies [BMB14, BRJM05] indicate that there may be post-processing in the human brain that allows the color space to be larger than their color sensitivity threshold indicates. The placement of the genes makes it possible for women to have four different cones [JDBM10], though there is no indication that they have four dimensional color vision. It is also interesting to note that color blindness can occur over time. Baraas et al. [BCG⁺07] show that tritanopy can occur through dystrophy, which explains how people with this disability are affected to varying degrees.

There have been studies regarding perception differences based on sex. Women pay more attention to the red-green axis and men more to the brightness [BKJ04]. Women also have less saturation loss in the periphery [MPMP12]. There also appear to be sex differences when dealing with unique hues [Kue01] and naming monochromatic lights [AGFC12]. A more in-depth analysis of sex differences related not only to color was written by Vanston et al. [VS17]. As the color vision genes are connected to the X chromosome, women are capable of being heterozygous (two different versions (alleles) of a gene). Commonly, this is believed to not affect their color perception. However, studies from 2006 [HMPJ06] by Hood et al. and 2009 [BK09] by Bimler et al. indicate that heterozygous women do suffer from a compressed color space, but not as strongly as homozygous individuals. The compression for women who are deutan heterozygous appears to be larger than protan heterozygous.

2.4 Color Maps

Based on a state of the art report by Zhou et al. [ZH16], color map generation can be divided into four categories: procedural, user study based, rule based, and data driven. In the first case, attempts are made to create color maps that are applicable to different data sets. Perceptual color maps based on color spaces such as CIELab are an example. There are some rules [LH92] that are recommended when generating a color map. Order, uniformity and representative distance should be taken into account, and the map should not create false boundaries. Ware et al. [War88] recommend using the rainbow color map for a good metric comprehension, and a gray-scale color map for shape comprehension.

User study based color maps are created through user interaction. In a method created by Kindlmann et al. [KRC02b], the user must determine when two faces are equiluminant for various different colors, which is then used to generate an isoluminant one dimensional color map. Rule based color maps are described [ZH16] as being based on rules regarding human perception in order to adapt the color map to both task and data. The last group, data driven color maps, create color maps based on the underlying data. Of course, there is significant overlap between these groups.

Bernard et al. wrote a survey of 2D Colormaps [BSM⁺15]. They list different criteria for 2D colormaps, as well as analytical tasks. Depending on the task, the requirements are different. This means that an adaptation of the color map for people with CVD, such as with Paper A, must take these criteria into account. The criteria they list are

- **Color exploitation:** Use as many colors as possible, allowing for large contrast.
- **Perceptual linearity:** Distance in the data range should be reflected in the color distance.
- **Equal visual importance:** There is no prior knowledge, or separate data groups are equally important.
- **Separation or separability:** Differences between values should be visually distinct.
- **Rows and columns:** Color characteristics should be kept separate. For example, a variable that changes in one dimension would only change in hue or lightness.
- **Background sensitivity:** Background data, e.g., known absence of data, should be visually distinct.
- **Device independence:** Allow portability between mediums.
- **Ease of implementation:** Construction of the color map should be easy and understandable.

They also evaluate and issue a guideline for task based choices of 2D color maps, as well as recommended color maps. Their guideline is based on six different tasks, three elementary and three synoptic:

- **Elementary localization and identification:** Identify the color of a data point, or localize an item based on its color.
- **Equal visual importance:** Avoid steering attention.
- **Elementary comparison:** Identify and localize the relations between data.
- **Synoptic localization:** Detect and localize a known group or class.

- **Synoptic identification:** Identify unknown groups or classes.
- **Synoptic comparison and synoptic relation:** Comparison of elements within and between sets.

Steiger et al. [SBM⁺15] proposed an application for interactively evaluating and creating a 2D color map. The application gives feedback on characteristics of the color map, such as the “just noticeable difference” (JND). It also allows the user to inspect the change in JND across the map via different plots.

When choosing color for data, the visualization goal as well as the data type must be taken into consideration. Ihaka [Iha03] gives some guidelines on clearly visible, harmonious and appealing colors. They give the following three guidelines:

1. Avoid high chroma colors as they tend to be distracting and cause afterimages.
2. Areas of similar size should have similar lightness.
3. Colors should be easy to distinguish if group membership is important.

In general, they recommend using perceptual color spaces such as CIELab. Wang et al. [WGM⁺08] establish a framework based on 13 different rules. Their framework also takes into account the mixing of colors, as can happen in volume visualization. Other guides are given by Bergman et al. [BRT95] and Healey [Hea96]. Furthermore, using semantically useful colors (e.g., blue for water) can speed up the time taken to find information [LFK⁺13]. In the case for hierarchical data, an approach that constructs a tree in HCL space (a polar coordinate version of CIELab) was proposed by Tennekes et al. [TdJ14], where the root of the tree is grey, and the tree grows circular outwards with increasing saturation.

Colorbrewer [HB03] is a website that lists different color schemes. A user can interact with and choose an appropriate color map. “iwanthue” [Med16] is a website where the user can specify the number of clusters and their size, and the website divides up the color space into according sections. However, these color maps do not take the visibility of the items or classes in the visualization into account. Aspects of the data, such as size, need to be taken into account, as small objects are only noticeable if they visually stand out. Lee et al. [LSS13] take the size into account and adapt the colors accordingly. Using a map of North America as an example, they compare their results to Colorbrewer, and alter the luminance and saturation of colors based on visual spatial frequency, making the larger areas less saturated and lighter, while the smaller areas are more saturated and darker.

2.5 Recoloring for CVD

When adapting visualizations and images to people with color blindness, there are several different ways to approach it, partially based on what the topic is. There are two basic cases: Recoloring of non-scientific images and recoloring of data and scientific images. In the first case it is not the goal to preserve the information and relation between information, while this is the case in the second one.

Recoloring data and scientific images can be done in several ways. First is to use something else than colors, such as patterns [SMO⁺13]. The second is to alter the images directly, and the third is to alter the color maps, either through a simulation of color blindness or by adapting the color map to the user through different measurements. While the last two achieve similar results, the difference is in the steps. In the first case, the image is generated and the colors are altered. In the second, the data is mapped directly to a different color.

There are different approaches to image recoloring, and speed can be an issue as well. Even if an algorithm is very accurate and effective, it has very little use if it is not fast enough. Below are sections on recoloring methods. First, different approaches for image recoloring will be described and examples given. Second, there will be a section for image recoloring methods that aim at being quick or even in real-time. Third, different methods for altering color maps are described.

2.5.1 Basic Image Recoloring Methods

One approach to recoloring images involves switching to grey-scale. This involves mapping the color to some grey value, based not only on luminance, but also color and saturation (chroma). An example of this is Color2Gray by Gooch et al. [GOTG05]. This method has three steps. First the RGB values are converted into CIELab. Then the differences between pixels creates a target color for each pixel. As it is not possible for each pixel to receive its target color, the algorithm ends with an optimization through gradient descent.

In 2005 Rasche et al. [RGW05] proposed a method for re-coloring images for gamuts of lower dimension. While this was done foremost for converting color images to greyscale—as color images were generally converted to greyscale when printed in a journal—they showed that this method can also be easily extended and applied for people with a form of color blindness. However, the method suffered from artifacts from compression and a lack of performance. In 2008 Bao et al. [BWMG08] improved this work by splitting up the color plane for people with certain color deficiencies and by mapping the colors from the original colors onto one part of the plane based on their RGB values. In the case of protanopes, the plane is divided into parts, one from black to white, the other from black to yellow. Mapping the original color to one of these two is by altering the blue value of the RGB color. Based on the type of color vision deficiency, a different component is altered. This method gives a considerable speedup. They show that their method

is approximately three orders of magnitude faster. In 2009 Huang et al. [HCJW09] proposed a method which uses CIELab. The colors are transformed into CIELab space, where they are analyzed using a Gaussian Mixture Model. Afterwards, the means of the Gaussians are translated based on the type of color vision deficiencies and used to interpolate the new colors. However, it takes several seconds to process a 300×300 pixel image. In 2011 Park et al. [PCH11] designed a method based on color confusion lines. They make a histogram of the colors of the image in HCL space (a polar coordinate version of CIELab) and group hues that are preset. A seed point is set and values around it are added until the maximum allowed difference in hues present in the picture is reached. Then the algorithm checks if groups are on the same confusion line and moves them to a free area (i.e., not occupied in histogram) if this is the case. They implemented this as an app for Android. In 2011, Orii et al. [OKMK14] proposed a method based on self organizing maps [KSH01]. The colors from the image are extracted. In this case, the image has sections that have the same color, such as a map of different counties. The self organizing maps algorithm is used to create a graph with the colors as vertices. Vertices that are similar in CIELab space are merged together. Then non-adjacent vertices are compared. If they have a similar color, then they are moved away from each other. This is repeated until there are no more confusing colors. In 2016, an approach based on similarity maps was proposed by Tennenholtz et al [TZ16]. In this method, the original image is taken and a simulation of what a person with color vision deficiency sees is created. They then create a similarity map for both images and compare them. This marks the positions where a person with color vision deficiency does not see any difference compared to a person with normal color vision. They then alter the RGB values of these positions, trying to reach similar values of difference, while incorporating a cost function that penalizes straying from the original image.

2.5.2 Efficient and Realtime Recoloring Methods

Kuhn et al. [KOF08] proposed in 2008 a method for converting an image of size 1024×768 in 0.27 seconds using a mass-spring model. The perceived color space of a person with dichromacy is approximated using a plane in CIELab space. Each color in the image receives a point in CIELab space and a spring is placed between each color with the resting distance being the Euclidean distance in CIELab space. The points are then placed on the two half-planes in the position the person with dichromacy perceives them. Afterwards, a physical simulation is run on the planes. Because the resting distances of the springs depend on the Euclidean distance between the original points in CIELab space, the simulation attempts to achieve similar perceived distances on the two half planes. The idea of moving the values onto a plane was expanded on by Liu et al. [LWY⁺09]. Liu et al. approximate the gamut using 2 half planes as well, and rotate the colors onto them. However, they expand it to video by using frame coherency. Normally, the colors are split up and one part is rotated onto one of the half-planes, while the other points are rotated onto the other. However, this can mean that the split is different for two different frames. Liu et al. measure the difference between frames. If the difference is low, then the colors are split similarly. Liu et al. achieve this in real time. Machado et al. [MO10]

demonstrated a similar real-time technique that also allows for additional color contrast enhancement. There has also been an approach to real-time recoloring of videos that take frame-consistency into account by Jeong et al. [JKW⁺12].

2.5.3 Color Map Alteration

An alternative to altering a picture that already exists is to alter the color map itself, i.e., to map the data or objects to different colors. This is not limited to complex data. In 1999, Viénot et al. [VBM99] described a set of color schemes designed for graphics applications running on Windows or Macintosh operating systems. However, they did not show a method to replace the colors themselves. Rather, they mapped 256 colors commonly used in applications to the version seen by protanopes and deuteranopes. This allows the designer to check how well a person with color vision deficiency can understand the information encoded through color. However, alternative approaches need to be taken for more complex data. One-dimensional data can be sorted using a simple optimized color transfer function. In the case of volume data, the color may depend on view point, and a method for direct volume rendering was proposed in 2011 by Chen et al. [CCB11]. When converting from normal colors, there is an attempt to try and maintain the original color scheme, i.e., the colors should change as little as possible. Unlike normal images such as photos, in volume rendering, the visible colors depend on the current perspective. The approach by Chen et al. takes this into account and attempts to optimize a linear system. This results in a transfer function from a data value of a voxel to a color, so the calculation only has to be done once. The calculation time depends on the parameters used, and their test cases were solved from between a fraction of a second to just over 17 seconds.

In the case of multidimensional data, several additional problems occur for people with color vision deficiency. First, as with normal color vision, the data itself must be reduced in dimensions in order to be visualized. During this reduction, data clusters may inadvertently be separated or created. As such, projections onto a color map must attempt to maintain the structure of the data, often aided by projecting onto perceptual color maps. As described by Mittelstädt et al. [MBS⁺14], the tasks generally fall into one of two categories: distinction between groups, and looking up groups. Generally, the projection onto the two-dimensional color space is achieved by optimizing for a cost function. This raises the second problem which applies to people with CVD: the creation of such a perceptual color map can be difficult. Not only is the gamut of the viewer reduced, but the Gamut of a computer monitor will reduce this even further.

However, the color map may need to be adapted to the person individually. For example, the monitor may be not calibrated, or the person's color vision deficiency may exist on a spectrum as is the case for anomalous trichromats. Usually, the user is shown different colors and asked if he can see any difference. One such way is by showing three colors in a line one next to the other and asking which of two colors is the most similar to the third [TKM⁺14]. Alternatively, a grid of circles is shown and the user is asked if he could see any difference, or a Landolt C is shown and the user is asked in which direction the

opening is [FG12]. However, few attempts have been made to alter a color map. Gresh et al. [Gre10] measured the color differentiation for a one dimensional color map and altered it accordingly. This was done by showing the user a color bar and repeatedly asking the user to find a square with a changed color within a short period of time.

2.6 Biomolecular Visualization

There are numerous different physical properties that can be visualized in molecular biology [KKL⁺15]. However, the technique elaborated in Paper B uses the van der Waals (vdW) surface representation [Ric77]. The vdW surface is a space-filling model where the radius of the atom spheres is proportional to the van der Waals radius. This surface shows the molecular volume, that is, it illustrates the spatial volume the molecule occupies. The van der Waals surface of a molecule is an abstract representation or model of that molecule, illustrating where, in very rough terms, a surface might reside for the molecule based on the hard cutoffs of van der Waals radii for individual atoms. It represents a surface through which the molecule might be conceived as interacting with other molecules. While higher-level structures such as α -helices can be shown using the vdW method, there are alternatives, such as the commonly used cartoon representation introduced by Richardson [Ric81]. In the case of large scenes, it may not be possible to represent every atom by a highly-tesselated sphere. In this case, impostor rendering is used. Further information on molecular visualization can be found in the state of the art report by Kozlikova et al. [KKL⁺15]. The work in this thesis is based on the cellView system of Le Muzic et al. [LMAPV15].

2.7 Flicker Physiology

Flicker can be described as a sensation that is “evoked when intermittent light stimuli are present to the eye” [Dav12]. The frequency at which the flicker becomes a stable signal is called the “Critical Flicker Frequency” (CFF). Above this frequency, the perceived color and luminance of the flicker should be equal to the average color and luminance, which is called the Talbot law [F.R34]. However, this law does not appear to hold universally [SP98], [Gre15]. The exact causes for the deviations are not known, and further study is required. The CFF itself depends on several different factors: Fatigue [Dav95], Age [BCS07], Training [SNHW05], Background illumination [Dav12], Amplitude [Dav12], Luminance [Dav12], [HV33], Size [GH30], Color, with the B-cone being worse than the R and G cones [Boy92], and the G cone is slightly better than the R-cone [Tyl87]

Early attempts at measuring how luminance affects the CFF resulted in the Ferry-Porter Law, which states that CFF is

$$CFF = k(\log(L) - \log(L_0)) \quad (2.1)$$

where L is the luminance, L_0 the threshold luminance for detecting the light, and the constant k is typically around $12Hz$. Further research by Tyler [TH90] indicates that

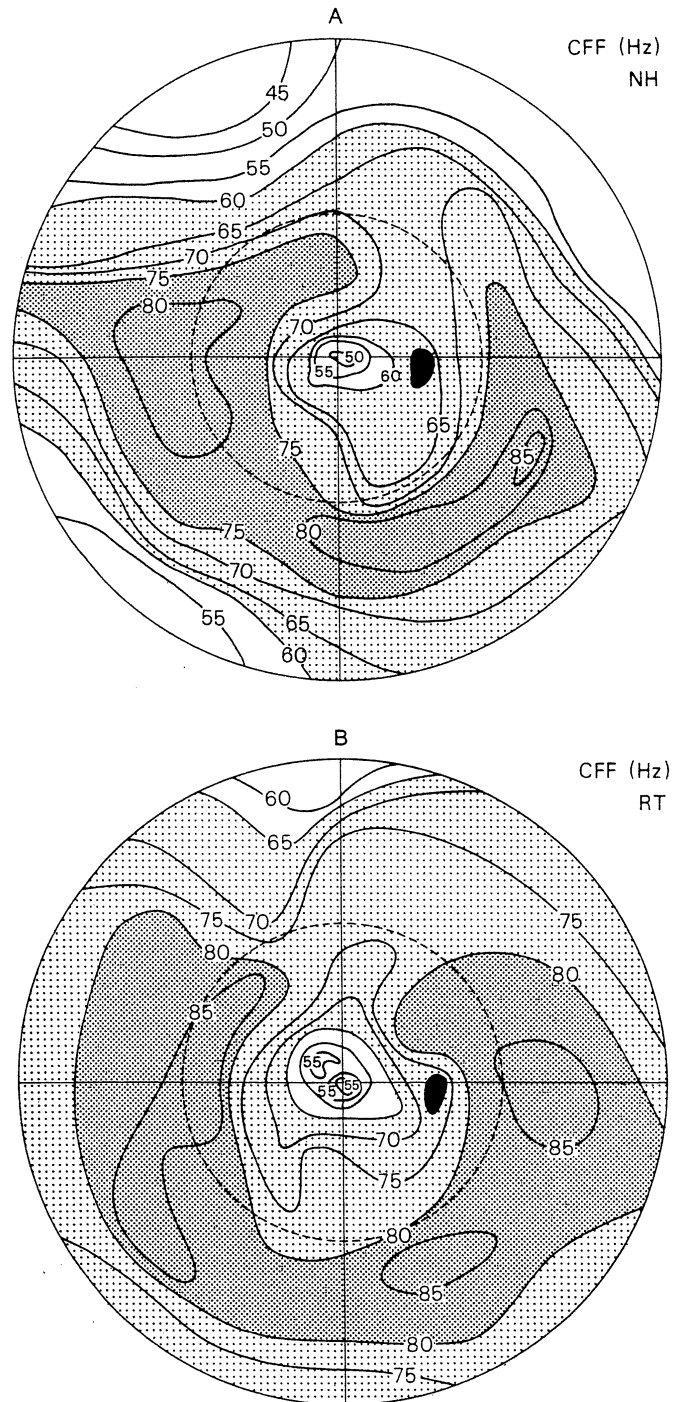


Figure 2.3: Figure 4 from “Analysis of visual modulation sensitivity. III. Meridional variations in peripheral flicker sensitivity” by Tyler [Tyl87]. The images shows measured critical flicker frequency sensitivities for two subjects. The results show that flicker sensitivity is higher in the near periphery (approx 15°-45°) than in foveal or far peripheral vision. This is used in Paper D.

this is a good measure. The Granit-Harper Law [GH30] says that the CFF increases with the log of the size of the flickering area.

A more interesting effect is the detection of flicker based on the visual angle from the periphery. The eye has the most receptors in the fovea, and the number drops extremely quickly outside of it, quickly reaching a low plateau. This means that the spatial resolution in the periphery is low. In contrast, the temporal resolution is generally higher, though the distribution is not particularly simple. It is important to note that the rods and cones have different sensitivities to flicker [HV33]. As rods are effectively inactive during stronger illumination, which receptor cell is responsible for detecting flicker depends on the strength of the stimulus as well as the light/dark adaptation of the viewer. While the difference in temporal integration across the retina has been known for some time, the effects and distribution are still being measured [Fuk79, TH90, TH93], as well as the detection of flicker of CRT monitors [Far86, Sid97].

Tyler [Tyl87] measured the detection of flicker on different parts of the retina. Figure 2.3, taken from his paper, shows the distribution for the two people he measured, using light aimed at the R-cone. The results show that the temporal sensitivity increases from the fovea to approximately 35° degrees of eccentricity, at which point the sensitivity declines again, forming an inverted 'U'. This finding has also been measured by other researchers. Pogel et al. [PTCS06] measured flicker detection between the fovea and up to 35 degrees of eccentricity. Their measurements led to calculating the CFF in Hz as

$$f(L, E, d_e) = (0.24E + 10.5)(\log(L) + 1.39\log(d_e) - 0.0426E + 1.09) \quad (2.2)$$

where L is luminance, E is eccentricity of stimulus, d is stimulus diameter at eccentricity E . This means that the overall formula for CFF between 0 and 35 degrees of eccentricity is

$$f(L, E, d_e) = (\delta E + \alpha)(\log(L) + \gamma\log(d_e) - \sigma E + \beta) \quad (2.3)$$

2.8 Flicker Application

Because flicker is a very strong attractor, there have been several different attempts to use it. However, previous attempts have—for the most part—always been at lower frequencies. This is partially due to high frequency monitors not being widely spread. There have been several attempts using low frequency flicker [BWC03, WB04, HBW08, GBM07]. The frequencies and amplitudes were varied. The users responded that the flickering was either ineffective or disturbing. This is a common problem with lower frequency flicker. Because it is effective at grabbing attention, it is also very difficult to ignore. Therefore, attempts at removing flicker after the point of interest has been detected have been made. Waldner et al. designed a method using a decaying flicker [WLMB⁺14]. In this approach, the flicker starts out at a high amplitude and frequency. After a few seconds the amplitude and frequency decrease and eventually reach zero. This allows the user to detect the flicker easily at the onset, but afterwards the decay of the flicker decreases the

irritating effect. However, the parameters of the flicker remained a significant problem, as both the starting and decay parameters need to be adapted to the personal preferences as well as the visualization itself. A different approach was done using an eye tracker. Bailey et al. [BMC⁺12] created the “Subtle Gaze Direction”, where a part of the image is flickering. When the viewer looks towards that part, the eye-tracker registers this movement and consequently the area stops flickering. This method has been shown to be effective both for static images as well as for narrative guidance through artwork [MBS⁺12].

There has been little research in the realm of using high frequency flicker that is only detectable in the periphery. Mateescu et al. [MB14] attempted to use high frequency flicker at 50 Hz, but was unsuccessful. An image was presented to the user, and the users were instructed to determine whether the flicker was on the left or right side of the presented image. The flickering area had a lower contrast than the rest of the image. The authors did not detect any effect. However, they had a different setup compared to the paper in Section D. They used a 21-inch CRT monitor with a resolution of 1024×768 and a chin rest 70 cm away, while we use a 27 inch LCD monitor with a LED backlight and a resolution of 2560×1440 and a chin rest 50cm away, allowing for larger visual angles along with a significantly higher resolution. An idea for a machine to create flicker to attract eye gaze was created presented in a patent by [Eva92]. However, we are not aware of whether this device was tested.

Verbose Contribution

The focus of the research presented here is to take advantage of the abilities and weaknesses of the human visual system. These capabilities do not necessarily have to be in the context of healthy human vision, as is demonstrated in Paper A. Furthermore, the idea is to adapt to the underlying functionality of the visual system itself, rather than a basic input. This means that the methods described here are adaptable to a variety of situations, not just a specific data set or type of data. For example, Paper B was applied to a data set containing molecular and protein data, and improvements (Paper C) allow it to be applied to any hierarchically organized data set. Other approaches, such as in the case of paper D, can be used by any consumer grade hardware on an arbitrary data set or image.

Another aspect of the approach is the effect of subtlety due to the use of visual system weaknesses. For example, when an area is normally highlighted, the area itself is significantly altered, e.g., drawing a circle around an item of importance. This causes the signal created by the highlighting to constantly be present in the mind of the person. If a researcher is interrupted by a strong signal, such as in the case of low frequency flicker, then the researcher must actively ignore the highlighting, which reduces the smoothness of the interaction with the data set, especially if the signal persists while the user is looking at it. Previous attempts at subtly directing the gaze [MB16, KV06, VMFS11], do function, but run the risk of being ignored. The methods presented here attempt to remove relying on chance. By cloaking our visualization methods with perceptual weaknesses, we can successfully evade the disruptive issue, while still managing to draw the person's attention to relevant data. This was attempted in two different areas. In the first case, the data the person is interested in is not known. Here, we adapt the visualization subtly and if possible on the fly in such a way as to maximize discrimination. In the second case, we draw a person's attention to a point of interest in a cluttered image. Here the indication is given in such a way that it is detectable in the peripheral,

but not in the foveal vision, meaning the person is not distracted when looking at the point of interest.

There were two other considerations that were taken into account during this research. A large problem for visualization is the deployment of the technology. In order for something to be useful, it must be deployable in the necessary circumstances. Our visualization techniques are designed for general use. Therefore, they must run on consumer grade hardware. Other approaches, such as the subtle gaze direction by Bailey et al. [BMC⁺12] require an eyetracker. Our techniques work on standard consumer hardware. Furthermore, the calibration of the technique to a person or monitor should be easy to achieve. Otherwise, it will remain unused because it is too tedious to set up. As such, the main goals of our methods can be summarized in the following list:

- Uses the weaknesses and strengths of visual system to its advantage.
- Subtle, does not distract, and is not detectable when looked at.
- Runs on consumer grade hardware.
- Easy to deploy or calibrate.

Paper A involves warping a 2D color map. It uses the color vision deficiency to measure perceptual difficulties in different areas of the color map. After warping, the fact that the color map has been changed is not directly noticeable, as it is still fairly similar. The method is designed to be used on consumer grade hardware and improve calibration speed. Papers B and C have been shown to run on consumer grade hardware even for large datasets. No calibration is needed, but calibrating the monitor to a perceptual color space can improve performance. Furthermore, the user can determine parameters on the fly. A hierarchical data set is colored based on hierarchical connections and the current view. One of the examples is an HIV data set. When the user is zoomed out, the coloring is based on the visible proteins. If the user zooms in and wants to see secondary structures such as helices, then the coloring adapts to the number of secondary structures, as well as which proteins they belong to. User studies show that the alterations in color can be hard to detect. Paper D functions on any monitor with a high enough frequency (100-144Hz required). Flicker is generated in such a way that it is detectable in the peripheral view, but not in the foveal view. This means that it helps the user orient himself, but is not distracting when the user looks at the point of interest directly. Some calibration is needed, but it is possible to do so in a matter of minutes. Below are the individual papers, their contributions, and how they implemented the above considerations and goals in detail.

3.1 Paper A: Personalized 2D Color Maps

For people with CVD, the visualization must take the reduced color space into account. This decrease is not necessarily uniform across the monitor’s gamut. Therefore, one issue facing this community is the measurement of the degree of color space decrease in different parts of the monitor color space, as this requires a fair number of measurements. Another issue is the adaptation of the visualization to the reduced color space, as this may vary within a subgroup of the CVD community, for example the different types of anomalous trichromats. Previous methods required the user to compare two colors and answer whether or not he could tell the difference, occasionally within a time limit. The minimum distance is then searched for via a binary search. Given the number of samples required to measure the effect across the entire monitor color space, this can require a fair amount of time, while also leading to exhaustion and irritation, along with the consequences of inaccurate measurements. Altering the color space generally targeted only specific types of CVD, where the perceived color space could be estimated (See Section 2.5 for details). In these cases, the perceived color space makes up a plane in CIELab. The altering of the colors used in the visualization is done by projecting the colors onto this plane in some manner. Not only is this not applicable to all types of CVD, but on a monitor it can also be very limiting in terms of used machine color space, and will not be an efficient solution if the monitor has not been calibrated to a proper color space, such as CIELab. Therefore we created a new measuring method, as well as a method used to alter the color space.

Rather than asking the person to simply compare two different colors, we had the user sort a line of colors. For the user study we limited the color space to a plane in RGB space. We extracted lines from the plane and sampled colors along it. These were presented to the user who had to sort them. This method was based on the game “Blendoku”, which is fairly popular and engaging. Using this method we managed to reduce the required time significantly. Furthermore, while sorting lines has been used to determine the type of CVD a person has, this is the first approach to use it to determine the degree of lack of color discrimination in a specific area.

To alter the color space, rather than projecting down to a subspace, our approach attempted to warp the color space itself. Effectively, a grid was constructed on top of the color plane and warped; an example of this can be seen in Figure 3.1 In areas where the color discrimination is poor the color space is contracted, i.e., the color distance between two points increases. This allows the overall Gamut of the color map to be used, while adapting to the CVD of the person.

By using these two parts, we can adapt the visualization according to our desired goals. The alterations are subtle as they affect the construction of the color map, they can be used in any visualization that requires a color map, and they can be deployed on any consumer hardware/monitor. Furthermore, we have improved calibration time over previous attempts.

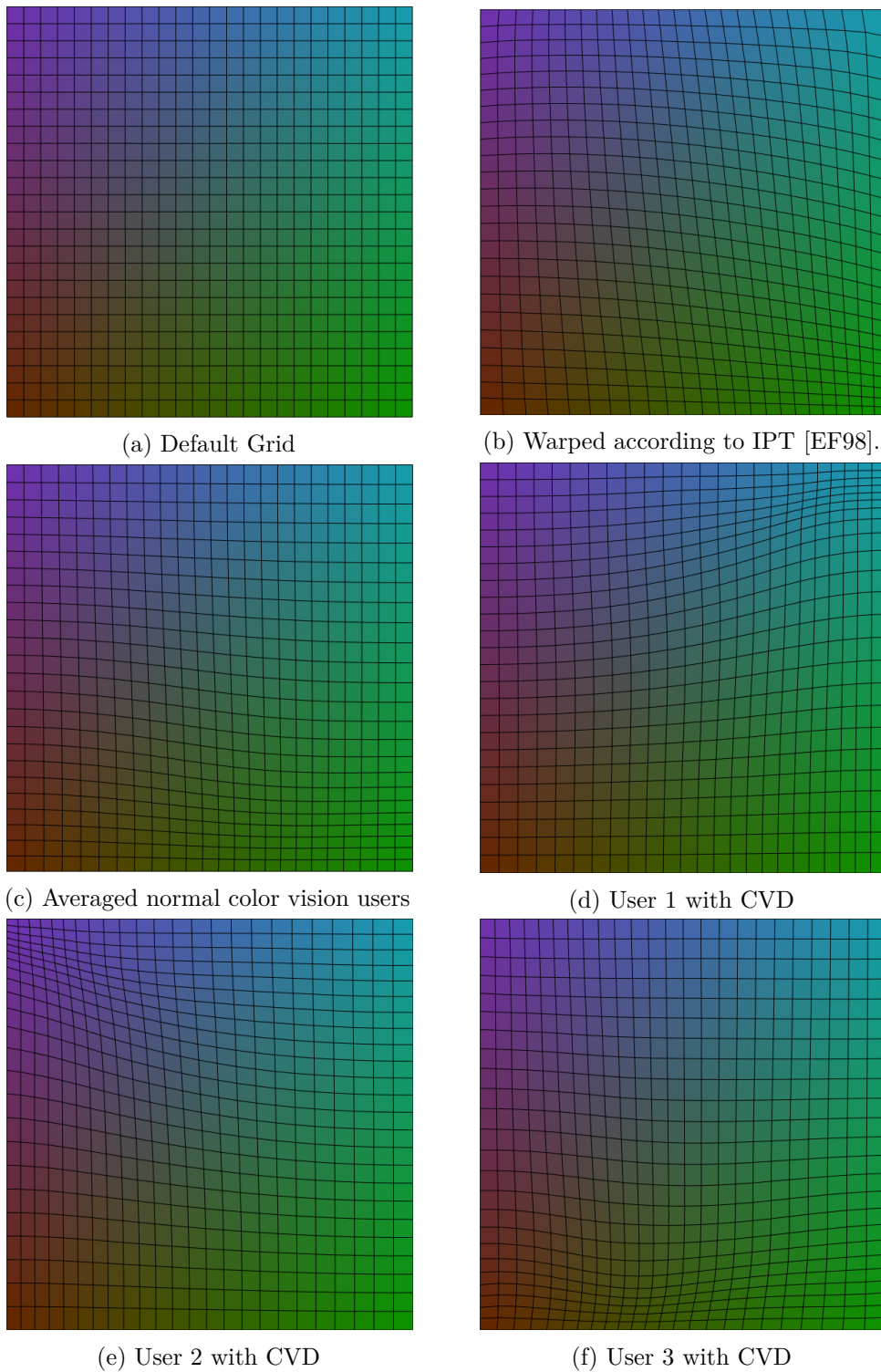


Figure 3.1: Warping of color map according to different test subjects and measurements

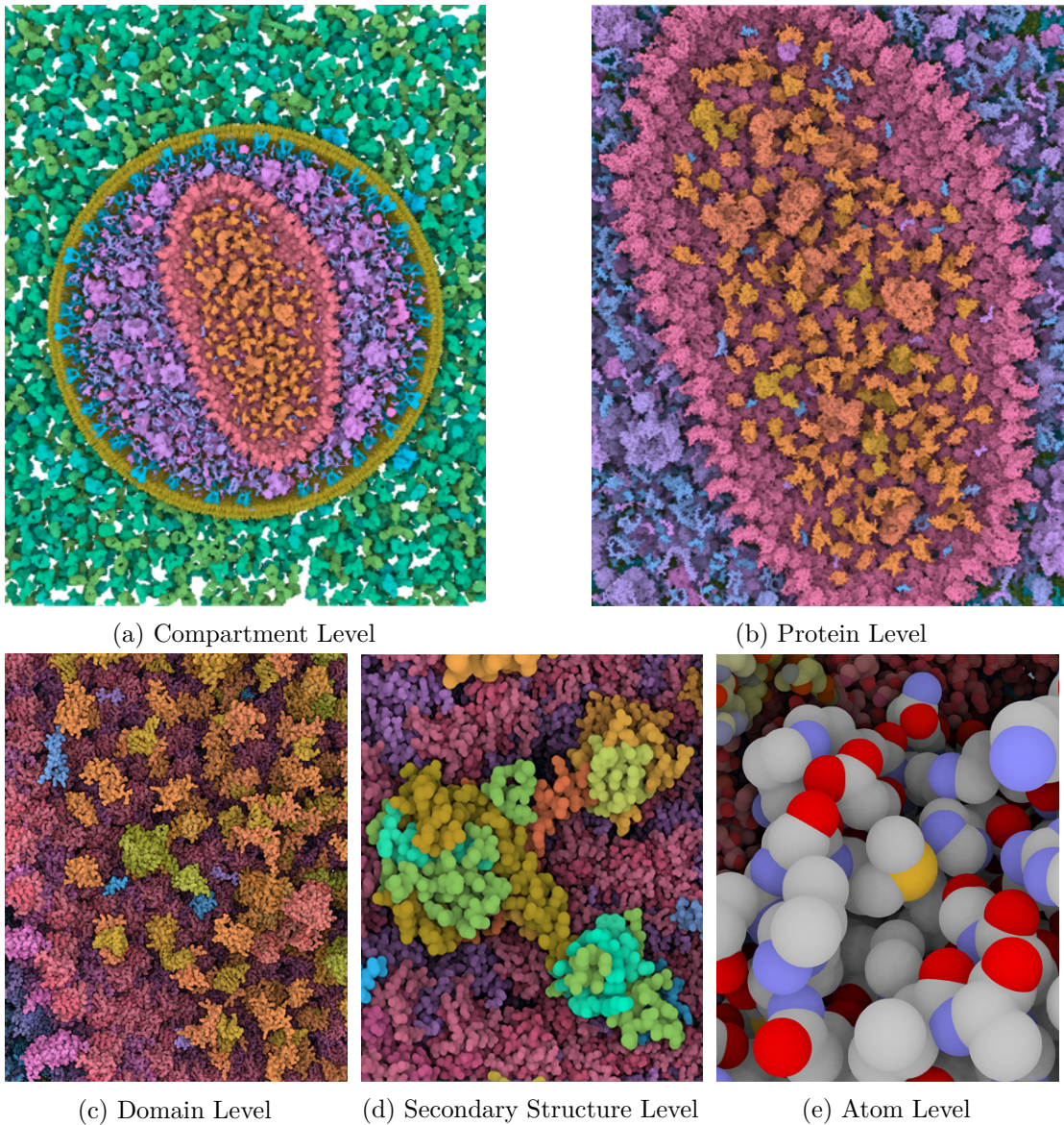


Figure 3.2: Chameleon (Paper B) coloring method for different levels of the HIV data set.

3.2 Paper B: Chameleon

Large quantities of data create significant problems regarding classification and detection. First, the data is hard to understand because of the quantity itself. Second, when attempting to visualize it, it reaches encoding limits. For example, naively giving every item a different color would lead to many being indistinguishable from each other. Therefore, the second paper concentrates on creating a hierarchical coloring scheme for

large data. The data was an HIV data set containing the structure of the HIV and protein information, from the protein level, down through the domain and secondary structure to the atomic level. An image of this can be seen in Figure 3.2. While the data is vast, it also contains hierarchical information, and it is on this aspect that we concentrated. It is important to note that the technique is designed for strictly hierarchical data.

A major issue with this data set is that we do not necessarily know what a person is interested in, meaning everything needs to be given an equal weight if possible. Furthermore, there is the issue of which level the person is interested in, e.g., the atomic or protein level. Therefore, the visualization must adapt not only to the number of currently visible items and not weigh one above the other, but must also adapt to the position or movement of the user. Furthermore, one of our stated goals is to not be distracting. Therefore, the adaptation must be subtle and smooth. We achieved this in the following manner. First we created a hue circle in CIE Lab, where each point on it has the same luminance. Each item that is visible from the current viewpoint receives a position on the hue circle. However, this simple approach would create too much noise when this scale will zoom out. To solve this, the hierarchical structure of the data is used. From top to bottom the data is structured as: Compartment of the virus, Protein, Domain, Secondary structure, Atom.

This means that we can construct a tree. If the HIV in its entirety is set to be the root, then the compartments would be the children at the first level, down to the atoms which are the leaves. Furthermore, users do not want to explore all levels at once. If they are looking at the protein level, then they are unlikely to be interested in the levels below that. Therefore, we take the distance between the camera viewpoint and the item into account. This allows us to regulate the number of items that receive a color. However, the connection between the parent and child nodes needs to be maintained, i.e., the placement of items on the hue circle must take their children/parents into account. We do this only in one direction, with the child taking the position of the parent into account. To describe this, there will first be a description without levels, which will then be expanded to multiple levels. If there is only one level, then all visible items are placed on the hue circle. If

the person moves the camera, then the number of visible items will change. If an item is no longer visible, then it is removed from the hue circle. If it reappears, it is placed back on the circle. In order to facilitate this, each point receives an original position,

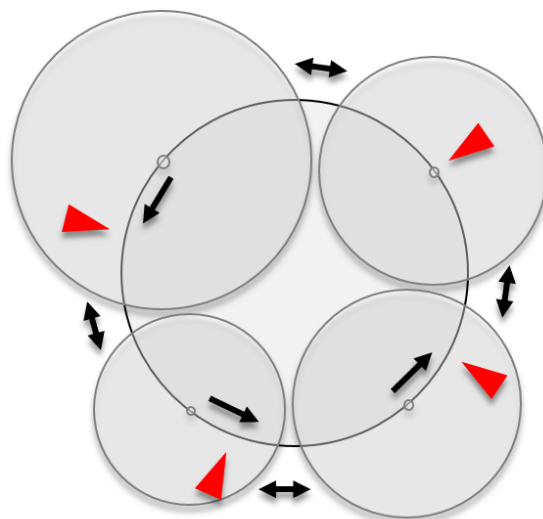


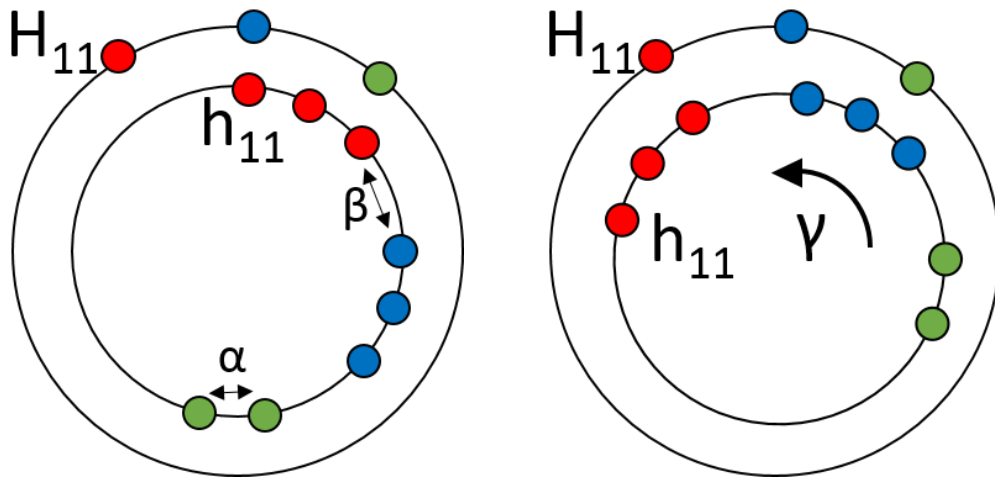
Figure 3.3: Diagram shows how the forces work. Red arrow indicates origin, black arrows indicate forces between objects and toward origin (parent). Here each circle indicates a group of children.

which corresponds to the position when all items are visible. This imposes an order on the points, which we can use for when an item disappears and reappears. Second, in order to maintain consistency during movement, we apply two forces. First is a force between an item's position on the hue circle towards its original position. Second is a force between items on the hue circle, keeping them away from each other, as shown in Figure 3.3. When hierarchy is introduced, the original position simply becomes the parent's position. In order to decrease the amount of calculations, we also group children together. However, this was not consistently applied to the HIV data set, because the 3D positioning of the data could be used to simplify lower-level color positions, which sped up the calculation. At the lower levels, an overlap of the hues between children from different nodes are allowed. For more information, please view Section 4.2 of Paper B.

Previous approaches attempted to create a static tree in the color space [TdJ14]. By having the items move on the hue circle and depend on the viewpoint, color becomes a reusable resource instead of a direct mapping, effectively changing the idea of what coloring is. In terms of fulfilling our stated goals, the method adapts to the visualization on the fly and does not need any specific hardware. There is no particular calibration needed, though a correctly color-calibrated monitor will help, which, given the right equipment, takes about 5-20 minutes. In terms of distraction or invasiveness, one would expect the constantly changing colors to be noticeable and problematic. However, in our user study, most people did not notice that the colors were changing, except at the lowest level, where a change to predefined colors occurred. Therefore, it is reasonable to say that the method is not distracting, in effect, taking advantage of a person's inattentive blindness to a degree.

3.3 Paper C: Cuttlefish

Chameleon (Paper B) was adapted to the HIV data set, and several simplifications were made in order to facilitate faster computation. This means the solution could not be applied well to an arbitrary hierarchical data set. An improvement to this was done through Paper C, "Cuttlefish". This paper solves two issues. First, it speeds up the calculation by simplifying it. In Paper B, the solution involves calculating forces and integrating them over time. Every time the image is rendered, the forces are calculated and a time-step is completed. While this works well for a few levels, when increasing the number of items and levels, the amount of calculation can become prohibitive. Cuttlefish uses the fact that all objects of a level are on the same hue circle. This allows the following simplified calculation: all the items are placed on the circle and then rotated by the same amount. The degree of rotation depends on the placement of the parents of the items. This also solves the second issue, the application to different data sets. Chameleon takes advantage of the 3D positions to allow for a simplification, but Cuttlefish can be applied to any hierarchical data set. How the rotation occurs and an example data set involving countries is shown in Figure 3.4 and Figure 3.5 respectively.



(a) Original placement of items on hue circle. β and α are user specified parameters that are used for determining angles between and inside groups respectively.

(b) Rotation of the children nodes by γ . This optimizes the distance between the children and the parent items by minimizing a cost function.

Figure 3.4: Placement of nodes on the hue circle. Parents are on outer circle, children are on inner. Color indicates group.



(a) Coloring of continents with their different subcontinents.

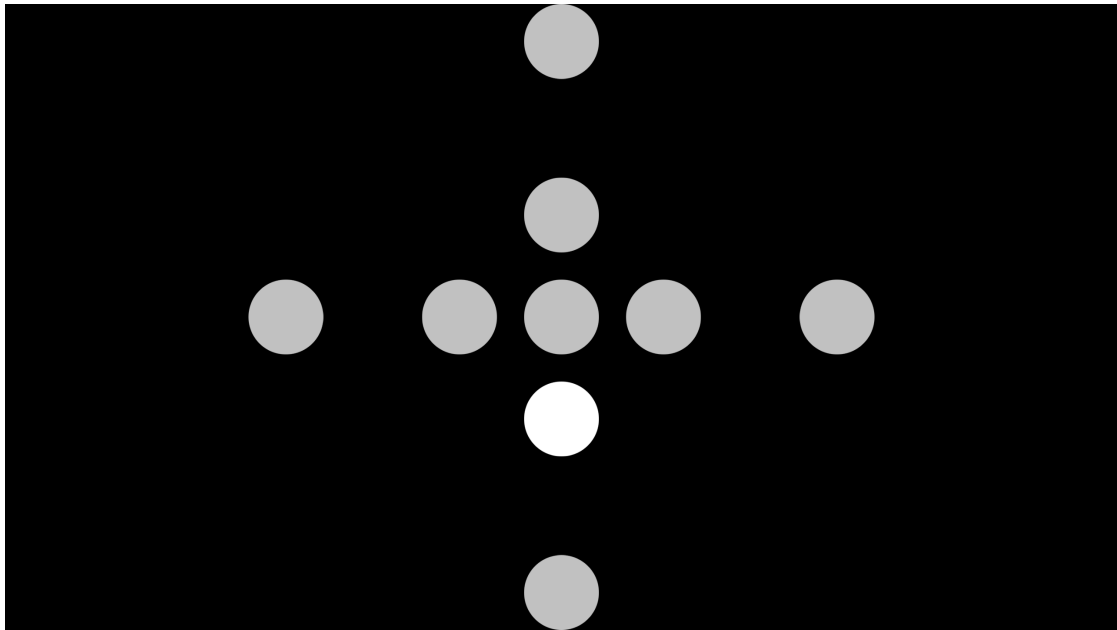
(b) Coloring of different American subcontinents.

Figure 3.5: Application of Cuttlefish on a zoomable treemap [Gan16]. The two images show data regarding population size and geographical location. The larger the rectangle, the larger the population. The color circles below the images show the assigned colors.

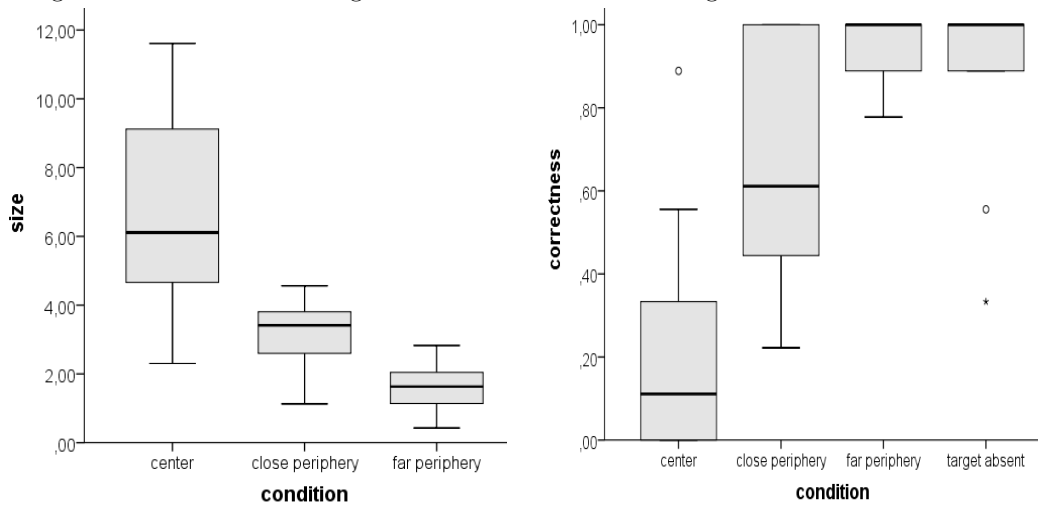
3.4 Paper D: Flicker Observer Effect

While the other papers concentrated on aiding visualization in data with an unknown target, this paper aims to guide the attention of the user through what we call the “Flicker Observer Effect”. As has been described in Section 2.7, the sensitivity to flicker varies across the retina, similar to an inverted 'U'. This allows a person to see flicker in the peripheral vision, but not in the foveal vision. The last paper uses this to its advantage to draw a person’s attention to a specific point.

In order to draw a person’s attention to an item in a cluttered image, the item must be highlighted in some manner, such as making it brighter, darker or circling it. Unfortunately, this may distort the item in some way, and needs to be actively ignored even while looking at the item itself, unless the user takes some action to remove the highlighting. A different approach involves making the item of interest flicker. This of course strongly alters the visual properties of the item and is highly distracting, so it should be coupled with a method to make the flicker decay over time or similar. However, it is possible that high frequency flicker can be detected in the periphery but not in the foveal vision. We used this effect to create a method for attracting attention in a way that can only be detected in the periphery. This means that the item would not be distorted when the user looks directly at it, nor would it be distracting. While numerous factors influence the detection of flicker, we concentrated on size, as luminance is given by the image, and frequency by the monitor. The other factors are either hard to control accurately and unnoticeably, or depend on the person himself. In a first stage we proved that this approach was possible, and in the second stage we applied it to a complex image, demonstrating its usability. The only calibration that was necessary for correcting the distortion of luminance because of the flicker was deducing a scaling factor for the non flickering parts of the image. Effectively, this shows we can use the “perceptual capabilities boundary” to our advantage, proving that we can go from working within the capabilities of the human perception to actually exploiting weaknesses and limits at a very low level of visual processing.



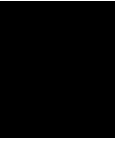
(a) Image shown to user for testing flicker detection. Here the brightened circle indicates flickering



(b) Boxplot of angle sizes required for the user to detect flicker. Bars indicate 3rd quartile, whiskers entire range

(c) Boxplot of flicker detection rate for users during test. Bars indicate 3rd quartile, whiskers entire range.

Figure 3.6: Testing and results from the Flicker Observer Effect paper (see Paper D). The top image shows the placement of circles which were used to measure the sensitivity to flicker. The bottom boxplots show the results.



Conclusion and Future Work

The research in this thesis describes methods that adapt or take advantage of the human visual capabilities. These methods use the weaknesses in human perception to adapt the visualization to the person without him/her noticing. Paper A adapts the visualization to the human color space while attempting to maximize the use of the computer monitor gamut. Paper B (Chameleon) and Paper C (Cuttlefish) show the current level of the visualization without disturbing the person when changing from one level to another through inattention blindness, displaying an optimal level of detail at a given time. The Flicker Observer Effect (Paper D) allows a person's vision to be guided without distorting the point of interest, simply by taking advantage of the difference in flicker sensitivity.

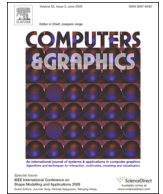
However, there is still a fair amount of work to be done. In the case of the CVD (Paper A) method, further research would investigate the accuracy of the line sorting method, as currently the research only indicates that it approximates the correct answer. A combination with other methods, as have been described in Section 2.5.3 may also be beneficial. In the case of Paper B (Chameleon) there are several things that could be approached. Paper B uses luminance to indicate lower level structures. For example, if a user is looking at the protein level, then the different domains will have different luminances. However, there is still little in the way of informing the user about two levels at once. Given the difficulties inherent in this task—not overloading the person with information, as well as remaining clear—an approach that takes advantage of the weaknesses of the visual system may be in order. Even with the improvements shown Paper C, no research has been done into any data structure, hierarchical or otherwise, containing a cycle or connections between items on the same level. For example, if there were a connection between items A and B from different parents, or a connection from A to B's parent, but not from B to A's. In other words, there is currently no method for showing connections or relations between items that are on the same level or a non-strictly

hierarchical data set. A form of similarity visualization would need to be incorporated into the system.

In the case of the Flicker Observer Effect (Paper D), issues still exist within the method. First, the flickering reduces the usable gamut of the monitor, effectively reducing the brightness and contrast, because the flicker in the study was between the original color and black. This flickering method can be further researched in two directions. First, a method for calibrating a monitor for a person for different images is necessary. The study only looked at the calibration of one image. There is currently no method for calibrating several images and then interpolating the necessary adaptations for an unknown image. Second, it is important to increase the usable gamut of a monitor. This would need to be achieved by flickering between two different colors, neither of which are black. This would also need to be tied into the method for interpolating the correct flicker settings for a new image. Furthermore, it might be desirable to extend this interpolation to people with CVD, as well as uncalibrated monitors. Using the flicker in images that are being viewed by several people at once has also not been studied. While our research indicates that a setting can be used by multiple people, our participants were for the most part quite young. As has been described earlier, there are several factors that affect flicker, age being an important one. Methods necessary for attempting to create flicker in such a way that it can be used for different groups at the same time has not been researched, and is also a very difficult problem.

There are still numerous limitations to the human visual system, but we don't understand the system completely, meaning that more weaknesses will still be discovered. While the research here touches on several aspects of it, the visual system spans several levels. As described well by Franconeri [Fra13], the visual system is multi-layered and flows through two streams, the dorsal and ventral stream. This thesis only investigates a small subset of possibilities, and further research is necessary, both in order to investigate potential opportunities for using current and future knowledge of the visual system.

Personalized 2D Color Maps



Special Section on SCCG 2016

Personalized 2D color maps

 Nicholas Waldin^{a,*}, Matthias Bernhard^a, Peter Rautek^b, Ivan Viola^a
^a TU Wien, Austria^b KAUST, Saudi Arabia

ARTICLE INFO

Article history:

Received 14 February 2016

Received in revised form

17 May 2016

Accepted 10 June 2016

Available online 24 June 2016

Keywords:

Color

Perception

Color vision deficiency

ABSTRACT

2D color maps are often used to visually encode complex data characteristics such as heat or height. The comprehension of color maps in visualization is affected by the display (e.g., a monitor) and the perceptual abilities of the viewer. In this paper we present a novel method to measure a user's ability to distinguish colors of a two-dimensional color map on a given monitor. We show how to adapt the color map to the user and display to optimally compensate for the measured deficiencies. Furthermore, we improve user acceptance of the calibration procedure by transforming the calibration into a game. The user has to sort colors along a line in a 3D color space in a competitive fashion. The errors the user makes in sorting these lines are used to adapt the color map to his perceptual capabilities.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Color maps are commonly used to convey data properties in height or heat maps. In a one-dimensional case the data is mapped onto a line in a color space, for example, RGB, CIE Lab, or a single color that changes in intensity. In two dimensions, either a plane is taken in a color space, e.g. RGB, or each axis corresponds to a color and a data value's color is a linear combination of the axes. Comprehension of two-dimensional color maps can be curtailed if the perceptual distance between data values mapped onto the color map is not consistent with the distance between original data values [1]. Fortunately, for people with normal color vision, despite having large differences in cone ratios, color perception is fairly similar [2], so standard perceptually uniform color spaces can be used. However, for users with a form of color deficiency, such as anomalous trichromats or dichromats, conventional color maps can raise several issues. First, there is the obvious problem of being unable to distinguish certain colors, such as red and green in the case of red–green blindness. The second issue is a change in perceptual distance. Because the perceptual distance between colors that are near each other can be very short, and for people with color vision deficiencies even look the same, the user will have difficulty in comprehending the data with these colors. As a result, he/she may overlook important features. This problem can be addressed by personalizing color maps. Our method extracts lines of color gradients from the color map and takes samples along it. These samples are displayed to the user as a sequence of squares, ordered randomly. The user must then sort these squares to achieve a smooth color gradient. An example of this task is shown in Fig. 1b, and an

example of the sorting in Fig. 1a. Based on the errors the user makes, we personalize the color space by contracting and expanding areas of the color map where the user performs worse or better. Through this we can personalize any continuous color space.

2. Related work

Two-dimensional color maps are used in a large variety of tasks [4], and are also used with higher dimensional data with data being projected onto a two-dimensional color map. In this case, tasks fall generally into one of the two groups: either into identification and comparison of data points and clusters or lookup of classes and clusters. For people without color vision deficiencies often perceptually uniform color maps [1] are used. The choice of color map depends on the task, and there is research on how to choose the appropriate color map [5]. An area where color map alterations have been demonstrated as useful is the remapping of colors for people with color vision deficiencies, for example encoding colors as patterns [6] and completely altering the colors of the image, both for stylistic reasons [7,8], as well as scientific purposes [9,10], and in real-time [11]. A longer list of examples can be found in a recent survey [12]. There are also methods for calibrating monitors to people with color deficits in order to recolor images, such as [13,14], which will be touched in Section 7.3. However, the trend of personalizing visualization and color maps is gaining momentum. One approach uses human perception of faces to generate isoluminant color maps [15]. An image of a face is divided into two parts, one white and one black. The image is copied and the white and black areas reversed. The image and its copy are placed side by side and shown to the user, with the black areas set to a gray value, and the

* Corresponding author.

E-mail address: nwaldin@cg.tuwien.ac.at (N. Waldin).



Fig. 1. In (a) an illustration of a person sorting a line is shown. The top line is presented to the person. S/He moves individual squares in the line in an attempt to sort it. In the second line one square has been moved. In the last line all the squares have been correctly sorted. In (b) a screenshot of what is displayed in our method to the user in the center of the screen is shown. The black area between what is shown and the edge of the monitor has been cut away. The colors have been altered significantly for easier perception. (c) shows a screenshot of the game Blendoku [3] on medium difficulty. Menu, stage name, and similar have been removed. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

white areas to a color. Two images are used because, if the luminance is unequal, one of the faces will stand out more. The user alters the luminance of the colored area until he/she thinks the luminance of the colored area is the same as the luminance of the gray area. This is done repeatedly, each time with a different color. The settings are then used to create an isoluminant color bar. Another approach [16] shows a method whereby a one-dimensional color bar can be adapted to an individual's color perception by searching for small changes in the color bar. This method can adapt a color bar in 10 min using 15 sample points. At each sample point the color of the sample point is taken and copied to another position along the bar. This creates a dot along the bar, which the user must search for and find within 5 s. While this method can be used to adapt two-dimensional color maps, it does not scale well. If we expand this method to two dimensions, this would mean about 15^2 (225) points of interest. Furthermore, working in two dimensions increases the number of directions. Because the number of directions in two dimensions is infinite, it is imperative to select the least number of directions that can still give a useful result. In the case of an arbitrary plane, all axes of the color map and their combinations are of interest, because the change in color and perceptual distances along a combination may be very different from the changes along an axis of the plane. Therefore, both the x -axis, y -axis and the diagonal directions $[x, y] = [1, \pm 1]$ should be investigated, leading to a total of 4 directions. As a result, in this approach the amount of work is first squared (number of points) and then multiplied by 4, leading to an approximate testing time of 400 min, or 6 h 40 min. If we assume that their approach uses as many points as we have areas in our experiment (20), then their approach would use $10 \times (20/15) \times 4 = 53$ min. However, the user will require a break. If we assume that the user takes a 5 min break after every 10 min, then it will take a total of 70–75 min. Our work has a similar goal. However, we developed a method that scales better than two dimensions is less exhausting and more enjoyable. In contrast, our method can be completed in as little as 30 min, though some users take significantly longer. Furthermore, it is influenced by the game Blendoku [3], which has a huge player base, indicating that it is enjoyable. It is also not as exhausting. In the approach paper by Gresh [16], the user must constantly react within seconds. In our game the user progresses as fast as he/she wants.

3. Methodology

People dislike using software that adapts to the users because it takes time, effort, and is usually tedious and boring. Therefore we have drawn inspiration from the field of mobile gaming in order to overcome these issues. Our method is based on the game Blendoku, where

the player must sort colors. In this game the player is given a figure consisting of squares. He/She must move squares with different colors onto the fields of the figure, such that the colors change from one hue to another between two different squares on the game figure. In our method, we do not include a puzzle aspect, as we only wanted to focus on the ability to distinguish colors. As such, we greatly reduced the difference in colors between tiles. A screenshot of our version can be seen in Fig. 1b, and Blendoku in Fig. 1c. In our game, the user is asked to sort a sequence of eight squares, the first and last cannot be moved. Using a pilot study with different color maps and methods of creating sequences, we analyzed what measurements could be used. We measured the number of incorrectly sorted squares, time spent comparing two colors, time needed to sort a sequence, and the number of times squares were moved. Only the number of incorrectly sorted squares was useful, as no other measurement correlated with it. Therefore, the method for adapting the color map uses only the number of errors made, and it attempts to alter the color map so that the areas of the color map with higher error are shrunk and areas with low error are expanded, increasing and decreasing the perceptual distance between data values. In this section, we will describe the probing and evaluation method. All RGB values are between 0 and 1 and in the format $RGB = [R, G, B]$.

3.1. Forced choice statistics

Our method is essentially a yes-or-no forced choice method. In forced choice statistics, the user must choose between two options, in our case if a tile is to the right or left of another. The probability distribution of normal forced choice follows the curve in Fig. 2. When the user cannot determine which option to choose, he must guess. The threshold for noticing a difference is placed at 75%. See Mckee et al. [17] for details. In our case of a line, the function will not decline as quickly, because the user can use additional squares to compare. For example, he may not see the difference between squares 2 and 3 and squares 3 and 4, but may see a difference between 2 and 4. It needs to be pointed out that if a person can solve a line without making a mistake, for example because it is very easy, then our method cannot tell the degree of difficulty of perception in this area. Another aspect of this forced choice is the effect on an adaption of the color map based on measured user errors. If a color map is optimally adapted to a person, then the perceptual difficulty would be evenly distributed over the color map for this person, i.e., the probability of an error would be independent of the position on the color map. If we take the mean perceptual difficulty and alter all areas to have this difficulty, then we will have achieved an even distribution of perceptual difficulty. This is described in Section 3.5.

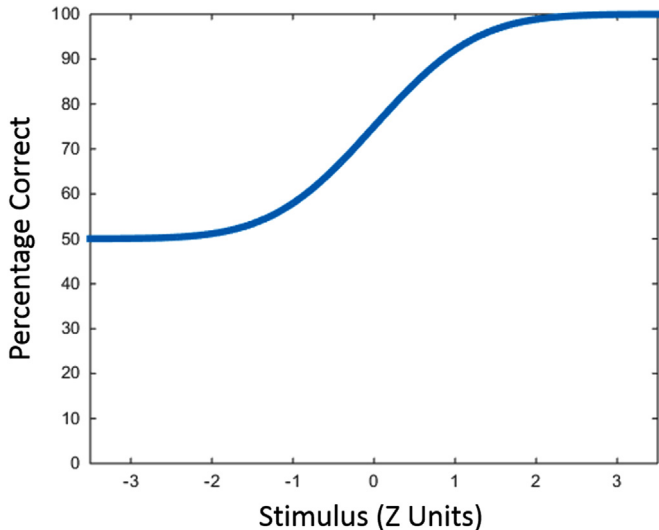


Fig. 2. Illustration of the curve generated by forced choice statistics when choosing which of the two items has more of some quality. Z units refer to the standard deviation, i.e., 2 means 2 standard deviations of the cumulative normal curve.

3.2. Game structure

The user has to sort a sequence of 8 squares, which we found to be adequate through a pilot study. All squares have different colors but are equally large. The squares at the ends cannot be moved, as they define the color gradient. The user can drag and move any square anywhere else in the sequence. This allows users to compare any two squares by overlapping one square with another. The sorting mechanism works in the same way as insertion sort. The user sorts the squares so the color changes smoothly from one end to the other. When he/she believes that the sequence is sorted correctly he/she presses the submit button. This is repeated until all sequences are sorted.

3.3. Selection of lines

The color map was divided up into a grid of rectangles. An equal number of lines, which have been generated at random across the color map, are randomly extracted from each rectangle. Points along the line are extracted to form a color gradient sequence. The change between squares in the color gradient sequence along the line is not necessarily the same for each step. This is due to line aliasing. Essentially, we rasterize the line in RGB space, as if the RGB space consisted of 256^3 voxels. However, the difference between two squares needs to be equivalent to two steps, e.g., $[0.0039, 0, 0] + [0, 0.0039, 0]$ ($0.0039 = 1/255$). Anything smaller, i.e., only changing one value, cannot be chosen because it leads to a loss of direction. The direction could change by 90° from one step to the next (e.g., one step $[0, 0, 0.0039]$, the next $[0, 0.0039, 0]$), which may confuse a user. This means a change along the line in RGB from one square to the next could be a change of $[0.0039, 0, 0.0039]$ the first time, and $[0, 0.0039, 0.0039]$ the next. Therefore, the error might not be simply along the line itself, but in the direction of a certain color. After 7 step combinations our line is created.

3.4. Evaluation of lines

The evaluation is based on the number of errors the user makes when sorting the lines. An error occurs when a square is in the wrong position relative to another square. This means that the incorrect placement of one square can lead to multiple errors. For example, if the square that should be in position 2 is placed in

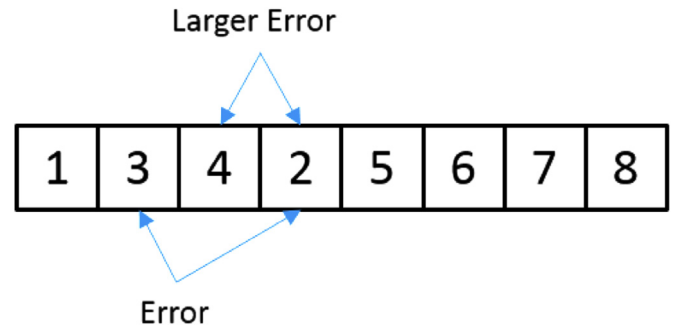


Fig. 3. Illustration of occurring errors in an incorrectly sorted line. The number refers to the correct index.

position 4, there are two errors, one for the incorrect comparison with square 3 and one with square 4 (which is now in position 3). An example can be seen in Fig. 3. The error consists of three components, the red, green and blue directions. For each line the errors in these three directions are added up. When a mistake is made, the error for this mistake is divided up into its RGB components and added to these three directions, similar to a histogram. This means that if a mistake consists of two squares and the difference is $[0.0039, 0.0039, 0]$, then 0.0039 will be added to R and G. There are two things to be considered when using the error for calculating the deformation: the effect of the error on the color map plane and the effect of the error on a specific direction. As described in Section 3.3, the line does not follow the plane precisely and this affects the measured error. Furthermore, our previously described binning of the errors creates a three-dimensional error vector for each line, which will not be parallel to the plane. Therefore, the error vector is projected onto the plane. The projection is orthogonal in order to take the angle between the error vector and the plane into account, which alters the magnitude of the error. This projected vector can then be used in the second step. We use PCA on the projected errors of each rectangle. This will give us a major and minor axis, which is similar to fitting ellipses, which is used in evaluations of color spaces [18]. However, PCA is more stable. By taking the dot product between a direction and the PCA axes, while considering their magnitude, the effect of the error on a direction can be calculated. The center of the group is calculated as the geometric center of the lines. This approximation of the error can be used to adapt the color map for the machine the test is done on. The sorting method allows noise to creep into the results, especially with only 7 lines per rectangle, which we used in our user study. To reduce the effects of noise, smoothing is performed before the PCA calculation. The neighboring rectangles are included in the PCA calculation, but to a lesser degree. The rectangle for which the PCA is being calculated has a weight equal to the number of neighbors. The neighboring rectangles have a weight of one. This reduces the effect of the noise and allows for a smooth transition across the color map. An example of this PCA analysis can be seen in Fig. 4, where the major and minor axes are displayed in the form of an ellipse for better visibility.

3.5. Morphing of color map

The color map is adapted to the measured errors. First the color map is divided up into a regular, rectangular grid for the purposes of deformation, in our user study 25×25 though finer grids can be used. The corners of the rectangles are defined as their position along the edges of the color map. If we take the bottom left as the origin, the right and up directions become the x and y coordinates. We use a mass-spring system in order to contract and expand areas across the color map to equalize the error, which is our goal as stated in Section 3.1. The vertices are the mass objects, and the edges are the springs. The mass-spring model can be described in

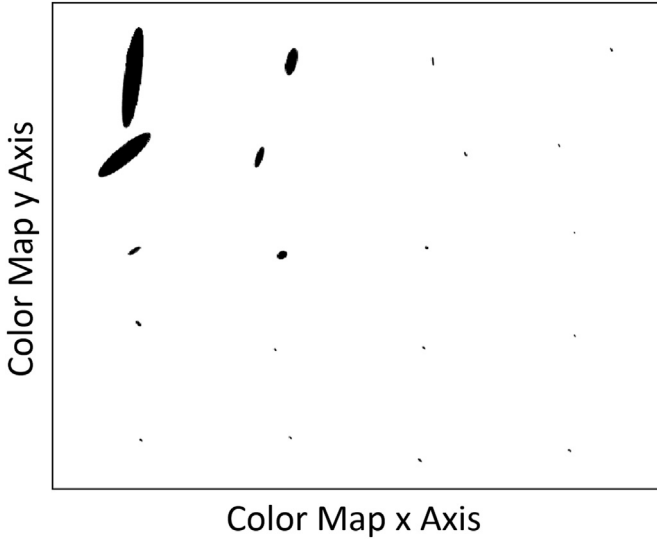


Fig. 4. Image of the color map with measured errors of each line group. The ellipses represent the actual errors from a test subject after the PCA step in the different rectangles, but have been scaled to be more visible.

the following manner. A spring h as a resting length L a constant k (here 1), and two ends. The spring applies a force equal to k times the dislocation from the resting length. The resting length corresponds inversely to the magnitude of the error. The greater the error, the shorter the resting length. We use squares instead of rectangles because we base the deformation on the size of the error and the resting length on the error relative to the mean, which allows us to compare the edges directly. This is simpler than incorporating the distance between vertices in RGB space, which in our case is taken care of when we project our deformation onto the color map because the ratio to the default state is kept. The springs are connected not only between the vertices along the x - and y -axes but also diagonally. A diagonal connection is required, otherwise artifacts can appear, such as neighboring springs along an axis forming a V shape. Furthermore, it allows us to cover more directions than just along the axes. Then, in order to morph the color map [Algorithm 1](#) is applied.

Algorithm 1. Calculate spring resting length.

- 1: $V_k \leftarrow error$ \\for the directions of all connected springs based on Eq. (1) for every vertex V_k
- 2: $V_k \leftarrow (V_k - V_{min}) / (V_{max} - V_{min})$
- 3: $V_k \leftarrow 1 - V_k$
- 4: $V_{mean} \leftarrow Mean$ of all V_k
- 5: $S_k \leftarrow (V_{k1} + V_{k2}) / 2$ \\where V_{k1} and V_{k2} are the vertices at the ends of the spring with resting length S_k
- 6: **if** S_k diagonal **then**
- 7: $S_k = S_k \cdot \sqrt{2}$
- 8: **end if**
- 9: Set the magnitude of the distances between vertices (i.e., along the edges of the squares) to the mean value.
- 10: Run mass spring simulation

The springs do not align exactly with the directions of error. For this reason, the weight of the errors depends not only on the distance from the spring, but also on the direction. The distance weight is based on a Gaussian kernel equivalent to a Gaussian with a standard deviation σ equal to the smallest distance between two line centers. A Gaussian kernel is chosen for the following reasons. First, a bilateral interpolation based on a quadrilateral mesh does not guarantee a C^1 continuity, i.e., a smooth derivative across an

edge. A kernel can achieve this, which effectively smooths abrupt changes that could occur at such an edge. Furthermore, there is an issue regarding the border, as we do not have any samples on it. A vertex on the border would not be influenced by a vertex diagonally across, even though this point might have important information, especially if there are no samples at the border vertex. Therefore, constructing a quadrilateral mesh is circular: in order to create a mesh to interpolate values, we must first interpolate values. Therefore, a kernel is used instead. A Gaussian kernel is used because a linear kernel behaves poorly for points far away. A linear kernel that reaches 0 at a distance of 10 would give a point at a distance of 8 twice the weight of a point at 9. A Gaussian kernel treats points that are far away from the center similarly and emphasizes points that are near the center. The error magnitude at a spring of direction \vec{d} , position p , and Gaussian weights G is calculated using the σ mentioned above. The sum of the weights is set to one. Using I as the set of all line centers and J_i as the set of PCA axes for each $i \in I$ with corresponding latent magnitude $m(\vec{j})$, the formula is as follows:

$$\sum_{i \in I} \frac{G(i, p)}{\sum_{i \in I} G(i, p)} \sum_{j \in J_i} m(\vec{j}) (\vec{j} \cdot \vec{d}) \quad (1)$$

3.6. Boundary conditions

The vertices on the boundary are allowed to move along the boundary edges only. The corners are fixed, and no vertex may be displaced around a corner. This has an effect on the distortions inside the color map. Because the mesh cannot freely contract, certain areas will have the same size, even if the resting length is different. Assume that we have two lines A and B parallel to the x -axis from one border to the other, but at a different y value and different but constant error magnitudes. Because the springs are pulling equally along the line, no vertices along A or B move. Therefore, both A and B are equally long.

4. User study

4.1. Users

Nine subjects with normal color vision and three subjects with a red-green color weakness participated; the particular type of weakness was not checked. Subjects were screened using Ishihara's tests for color blindness.

4.2. Calibration and setup

The user study took place in a darkened room, i.e., no lamps and with covered windows. The monitor was a standard white LED monitor that was calibrated to D65 color temperature and sRGB. No restrictions were set on the user regarding the distance to the monitor. This was allowed in order to provide similarity to a standard work space. The user was presented with a screen that contained the following: 8 squares in a sequence in the center of the screen, a submit button at the bottom, and a menu button in the top left. The background was almost black. Users complained that a completely black background caused strain on their eyes, so it was changed to RGB=[0.071, 0.071, 0.071]. Users had to solve a total of 140 sequences. When the submit button is pressed after each solution, the user was asked to rate the difficulty of that line from 1 to 10, 10 being the hardest. The sectors were traversed in the same predetermined, ordered manner for all participants. There was no indication that participants became exhausted during the experiment. The users were informed that they could take

as long as they needed and could take breaks if they wanted to, but were made aware that there were a total of 140 lines to solve.

4.3. Color map

The color map used in the user study has the endpoints [0.392, 0.157, 0] (reddish brown), [0.039, 0.588, 0] (green), [0.431, 0.196, 0.706] (purple), and [0.078, 0.235, 0.706] (cyan). An example can be seen in Fig. 5a. In all images in this paper containing the color map, these correspond to the bottom left, bottom right, top left, and top right corners respectively, and will be referred as such. These points were chosen to show that our method would work with varying luminance and color, and would not require a specially chosen color map. While a color map that only has variances in luminance or in hue can be chosen, the latter can cause difficulties, as it is impossible for the luminance to be equal for all points on the color map. If a sequence is extracted from such a color map, then the luminance along it varies. Essentially, it is aliasing. The computer would not be able to display each color at the same luminance due to the discrete nature of its color space. Therefore, the sequence would not appear ordered along a straight line to the user and can lead to confusion, as we noticed in a pilot study.

4.4. Color map division

As described more generally in Section 3.3, the color map was divided up into a grid of 20 rectangles, arrayed 5×4 . The longer axis had 5 along it. The user had to solve a total of 140 lines extracted from the color map; 7 from each rectangle at random. The lines were generated randomly across the color map. These values were chosen so the color map would be well covered.

5. Results

The subjects with normal color vision took 56 min on average to complete the study. The minimum time was 28 and the maximum 90 min with a standard deviation of 21.7 min. The subjects with red-green weakness took 55, 51, and 39 min respectively. See Table 1 for more information. There was no correlation between time taken and number of errors. All subjects with normal color vision had a similar result. The average deformation of people with normal color vision and a deformation based on the perceptually uniform IPT color space [19] as a comparison can be seen in Fig. 5. The distances in IPT space along the x - and y -axes can be seen in Fig. 8a and b respectively. The deformation of both the color map based on the IPT color space and that based on the averaged users with normal color vision contract in the bottom right (green) area. This means our method expands and shrinks in the correct areas. However, the IPT version contracts less vertically but

more horizontally. Unlike the deformation for users, the deformation based on the IPT color space alters the constants of the springs, rather than the resting lengths, which are set to 0. Because the distances in the color space are known, we can create a more exact deformation to try to preserve the perceptual distance. The constants are set to $1/distance_{IPT}$. This can be explained through a simple one-dimensional example. There are two springs. Along the first spring the perceptual distance is 1, along the second 2. If we set the constants to $1/distance_{IPT}$, then we have the constants $k_1 = 1$ and $k_2 = 0.5$. This means the spring with a perceptual distance of two will be twice as long.

The three subjects with red-green color weakness all had substantially different results. While the exact type of color deficiency of the users was not determined, this indicates that personalized color maps for people with a color weakness can be fairly unique, and so personalized color maps may be needed. Their distortions can be seen in Fig. 6. An coloring example based on one of these distortions can be seen in Fig. 7. Fig. 8c shows the difficulties reported by the users with normal color vision. The users considered the bottom right to be the most difficult, the same area where the most contraction occurs. In comparison, the distances in IPT space can be seen in Fig. 8a and b.

6. Validation study

A validation study under the same conditions was performed on the users with color deficiency. The users were shown an image of a curvy line on a monotone background. The colors were selected from default color map and the personalized color map.

6.1. Selected color

The color of the line and background were chosen in the following manner. A point on the color map where the most contraction

Table 1
Time to Solve (TtS) for different subjects.

Weakness	Median TtS (s)	Total TtS (min)
None	13.4	35
None	24.3	75
None	12.1	34
None	20.3	57
None	32	90
None	15.1	38
None	27.1	68
None	11.7	28
None	30	83
Red-Green	18.8	55
Red-Green	20.6	51
Red-Green	14.9	39

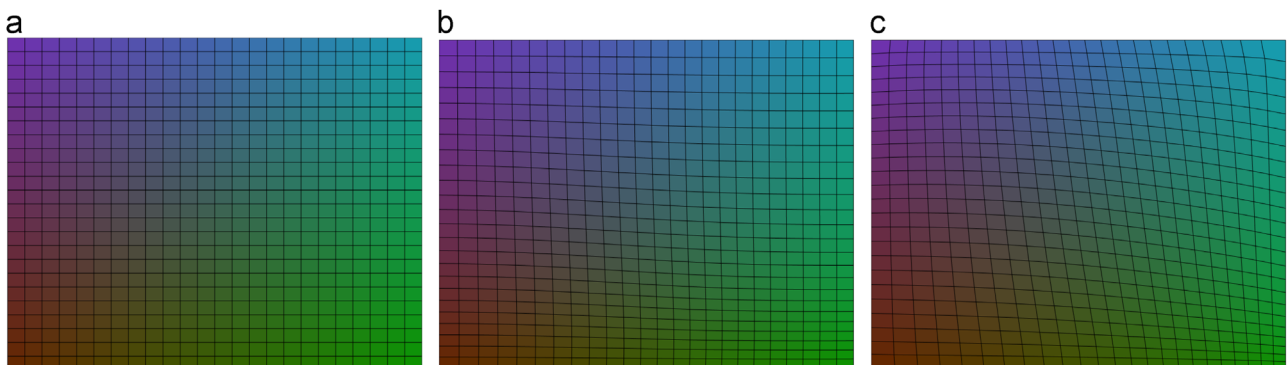


Fig. 5. These three images show the color map in an undeformed state (a), deformed by the averaged results of users with normal color vision (b), and deformed using the distances in IPT [18] color space (c). Both (b) and (c) contract and expand in the same areas, though to different degrees.

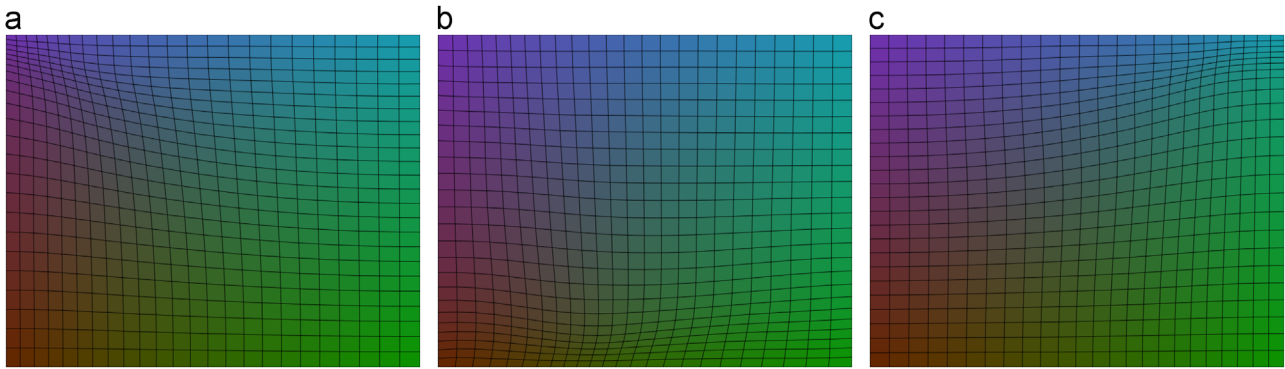


Fig. 6. Deformations of the color map for three people with red-green weakness.

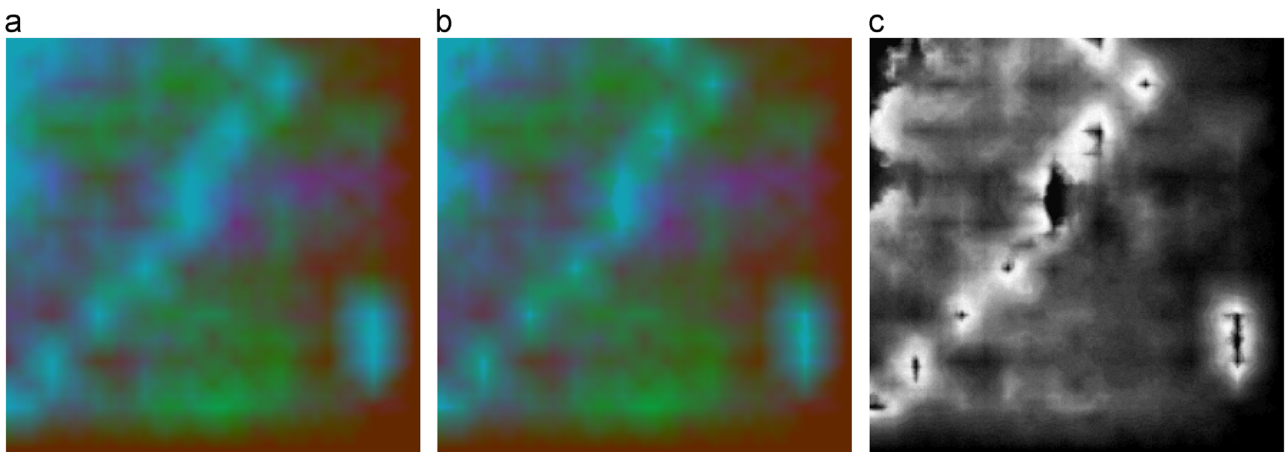


Fig. 7. The images show a map where each pixel has 2 data values. (a) is colored by the default color map, (b) according to Fig. 6c. (c) shows the normalized difference in value. In terms of distance in RGB space, the maximum difference is 0.123 and the mean 0.04. As stated before, axes in RGB space have a maximum value of 1. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

occurred was selected for each individual user as the background color. The color of the line was selected by taking a nearby point, i.e., the new point was offset from the original. The offsets were 0, 2, and 4 steps on the color map along the x - and y -axes and their combination (i.e., one diagonal direction). If viewed on the color map, the x direction is to the right, the y direction is downwards, and the diagonal direction is towards the bottom right.

6.2. Line variations

The line either went from the left to the right border of the image, or from the top to the bottom. The end points were not aligned and could be on independent, arbitrary points along the border. In order to avoid exhaustion, the number of line and color variations was minimized. Five lines were generated, 3 horizontal and 2 vertical. While there were a limited number of variants, the line had a random horizontal or vertical shift, depending on the endpoints. If the endpoints were on the right and left borders, the line was shifted randomly up or down, otherwise the line was shifted arbitrarily to the right or left. However, the shift was the same for each offset combination. For a given offset and line, the only difference would be the colors used.

6.3. Execution and results

These variations result in 80 images. The image was displayed for 3 s, after which the user had to say the direction of the line, or if he/she did not see a line. The number of cases where the user could not detect the line or mistook its direction was counted. Two

users mistook the direction of the line only once, and one user three times. Because these numbers are so low, these cases were counted as not seeing the line. The results can be seen in Fig. 9. As expected, the users all perform better with the adapted color map, though to varying degrees. If we take the colors from the adapted color map, then their distance on the original color map is 2–3 times higher, depending on the user.

7. Conclusion

The method of generating perceptually uniform color maps we have described can show where a person has difficulty in differentiating color, and it does so in a reasonable amount of time. It also has the benefit of being more enjoyable. The variation among persons with red-green color weakness indicates a need for personalized color spaces in such cases. Another application could be a grading of color maps based on the magnitude of errors. Furthermore, while it has not been tested, our method should be usable on a non-calibrated monitor under arbitrary lighting, similar to previous work [16].

7.1. Comparison to previous work

As mentioned before, a similar method was created by Gresh [16], but can be expected to require 75 min to complete. Our method can be completed in 30–40 min, though some users take significantly longer. Furthermore, it is influenced by the game Blendoku [3], which has a huge player base, indicating that it is enjoyable. It is also not as

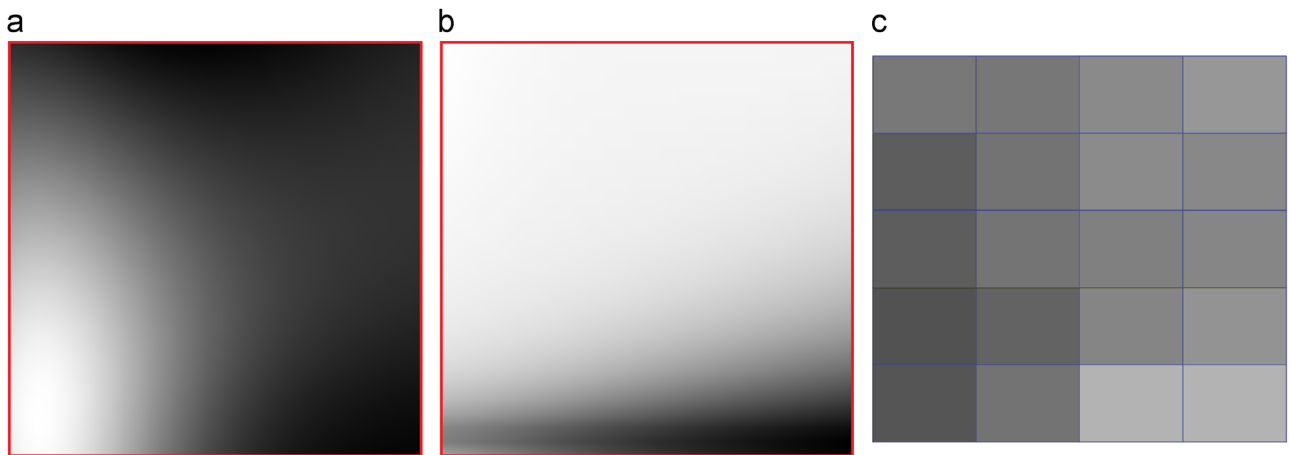


Fig. 8. Normalized perceptual distances of color map in IPT color space along x-axis (a) and y-axis (b). Darker indicates shorter distance. (c) Averaged normalized difficulties reported by users. Areas of higher difficulty are lighter.

Offset	0	2	4
0	X	■	■
2	■	■	■
4	■	■	■

Offset	0	2	4
0	X	■	■
2	■	■	□
4	■	□	□

Offset	0	2	4
0	X	■	■
2	■	■	■
4	■	■	■

Offset	0	2	4
0	X	□	□
2	□	■	□
4	□	■	□

Offset	0	2	4
0	X	■	□
2	■	■	□
4	□	□	□

Offset	0	2	4
0	X	□	□
2	■	□	□
4	□	□	□

Fig. 9. These tables show the capability of the individuals to detect curvy lines with offset color map values in an image. The top rows (a–c) are with unaltered color maps. The bottom row (d–f) with personalized color maps. The offsets were 2 or 4 along the x (towards the right) and y (downwards) axes. The lightness indicates the number of detections, with a total of 5 lines per cell. The darker the cell, the less lines were detected; black none, white all.

exhausting. In the paper by Gresh [16], the user must constantly react within seconds. In our game, he progresses as fast as he/she wants.

7.2. Limitations

The method still has several limitations. The color map must be fairly fine-grained, otherwise the steps between the individual squares become too large and the errors will be simply noise. Also, it is still a lengthy process. The fastest users took about 30 min. Furthermore, the samples are semi-random. As can be seen in the case for the users with color weakness, most of the contraction is in one area, wasting a lot of time. Furthermore, the method cannot detect if a user cannot distinguish two colors that are not next to each other, such as red and green.

7.3. Future work

Some of the limitations can be overcome. For example, adaptive sampling could reduce time and increase accuracy by increasing and decreasing sampling in areas with many or few errors. This would also help detect smaller areas where large contractions or expansions are required. Further research could reduce the number of squares in a line. The number affects the users' ability to sort them – more squares allow for more comparisons – as well as the degree of difficulty that can be measured. There are also some aspects that have not been researched. The study was done in a dark room. This makes it easier for the user to distinguish colors. It is unknown how well the method will perform in a normal working environment, as daylight can affect color perception [20]. Another question is consistency between people with color

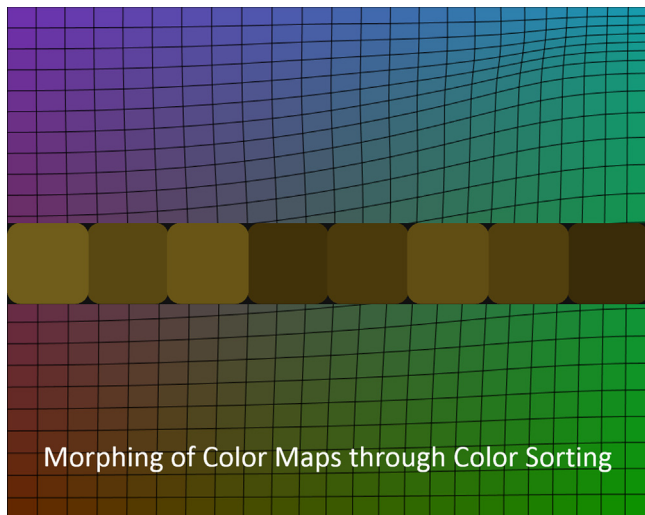
deficiencies. In our study, the users with a red-green color deficiency all had different results. It is unknown if this is due to them having different deficiencies, or such differences occur for the same deficiency. Lastly, it might be possible to tie in the monitor calibration methods shown in [13,14]. First, similar to the method in [13], confusion axes – axes along which the user has difficulty in seeing colors – could be used to speed up measurement and accuracy. Second, these methods try to measure a person's color vision across the monitor in order to use recoloring methods. They try to measure the distance at which a person can differentiate colors in certain areas of the color space. This measurement can be used, similar to the distortion of the color map using IPT space in Section 5 using the spring constants, to adapt any color map. However, as shown by the variation in areas where the color map contracted for different users, the accuracy and individualism are unknown.

Acknowledgments

This project has been funded by the Vienna Science and Technology Fund (WWTF) through project VRG11-010 and also supported by EC Marie Curie Career Integration Grant through project PCIG13-GA-2013-618680.

Appendix A. Supplementary material

The following are the supplementary data to this paper:
[Video S1](#)



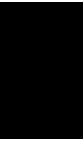
Video S1. This video provides a summary of the paper and describe the steps used to adapt a color map to an individual by sorting a series of squares taken from a line in the color map. A video clip is available online. Supplementary material related to this article can be found online at <http://dx.doi.org/10.1016/j.cag.2016.06.004>.

References

- [1] Mittelstädt S, Bernard J, Schreck T, Steiger M, Kohlhammer J, Keim DA. Revisiting perceptually optimized color mapping for high-dimensional data analysis. In: EuroVis – short papers, 2014. p. 91–5.
- [2] Hofer H, Carroll J, Neitz J, Neitz M, Williams DR. Organization of the human trichromatic cone mosaic. *J Neurosci: Off J Soc Neurosci* 2005;25(42):9669–79.
- [3] LonelyFew, Blendoku, 2015. URL (<http://www.blendoku.com/>).
- [4] Bernard J, Steiger M, Mittelstädt S, Thum S, Keim D, Kohlhammer J. A survey and task-based quality assessment of static 2d colormaps. In: Proceedings of SPIE, 2015. p. 93970M–93970M-16.
- [5] Steiger M, Bernard J, Mittelstädt S, Hutter M, Keim D, Thum S, et al. Explorative analysis of 2d color maps. In: Proceedings of WSCG, 2015. p. 151–60.
- [6] Sajadi B, Majumder A, Oliveira M, Schneider R, Raskar R. Using patterns to encode color information for dichromats. *IEEE Trans Vis Comput Graph* 2013;19(1):118–29.
- [7] Flatla DR, Reinecke K, Gutwin C, Gajos KZ. Sprweb: preserving subjective responses to website colour schemes through automatic recolouring. In: Proceedings of the SIGCHI conference on human factors in computing systems, 2013. p. 2069–78.
- [8] Kuhn G, Oliveira M, Fernandes LA. An efficient naturalness-preserving image-recoloring method for dichromats. *IEEE Trans Vis Comput Graph* 2008;14(6):1747–54.
- [9] Chen W, Chen W, Bao H. An efficient direct volume rendering approach for dichromats. *IEEE Trans Vis Comput Graph* 2011;17(12):2144–52.
- [10] Rasche K, Geist R, Westall J. Re-coloring images for gamuts of lower dimension. *Comput Graph Forum* 2005;24(3):423–32.
- [11] Machado GM, Oliveira MM. Real-time temporal-coherent color contrast enhancement for dichromats. In: Proceedings of the 12th Eurographics/IEEE – VGTC conference on visualization, 2010. p. 933–42.
- [12] Oliveira M. Towards more accessible visualizations for color-vision-deficient individuals. *Comput Sci Eng* 2013;15(5):80–7.
- [13] Flatla DR, Gutwin C. Improving calibration time and accuracy for situation-specific models of color differentiation. In: The proceedings of the 13th international ACM SIGACCESS conference on computers and accessibility, 2011. p. 195–202.
- [14] Flatla DR, Gutwin C. Situation-specific models of color differentiation. *ACM Trans Access Comput* 2012;4(3) 13:1–44.
- [15] Kindlmann G, Reinhard E, Creem S. Face-based luminance matching for perceptual colormap generation. In: Proceedings of the conference on visualization '02, 2002. p. 299–306.
- [16] Gresh DL. Self-corrected perceptual colormaps [Online], 2010. URL (<http://www.research.ibm.com/people/g/donnagresh/colormaps.pdf>).
- [17] McKee S, Klein S, Teller D. Statistical properties of forced-choice psychometric functions: implications of probit analysis. *Percept Psychophys* 1985;37(4):286–98.
- [18] Mahy M, Van Eycken L, Oosterlinck A. Evaluation of uniform color spaces developed after the adoption of cielab and cieluv. *Color Res Appl* 1994;19(2):105–21.
- [19] Ebner F, Fairchild MD. Development and testing of a color space (ipt) with improved hue uniformity. In: Color imaging conference, vol. 1998 (1), 2005. p. 8–13.
- [20] MacAdam DL. Visual sensitivities to color differences in daylight*. *J Opt Soc Am* 1942;32(5):247–74.

PAPER

B



**Chameleon: Dynamic Color
Mapping for Multi-Scale
Structural Biology Models**

Chameleon

Dynamic Color Mapping for Multi-Scale Structural Biology Models

Nicholas Waldin¹, Mathieu Le Muzic¹, Manuela Waldner¹, Eduard Gröller^{1,2}, David Goodsell³, Autin Ludovic³, Ivan Viola¹

¹TU Wien
²VRVis
³Scripps Research Institute

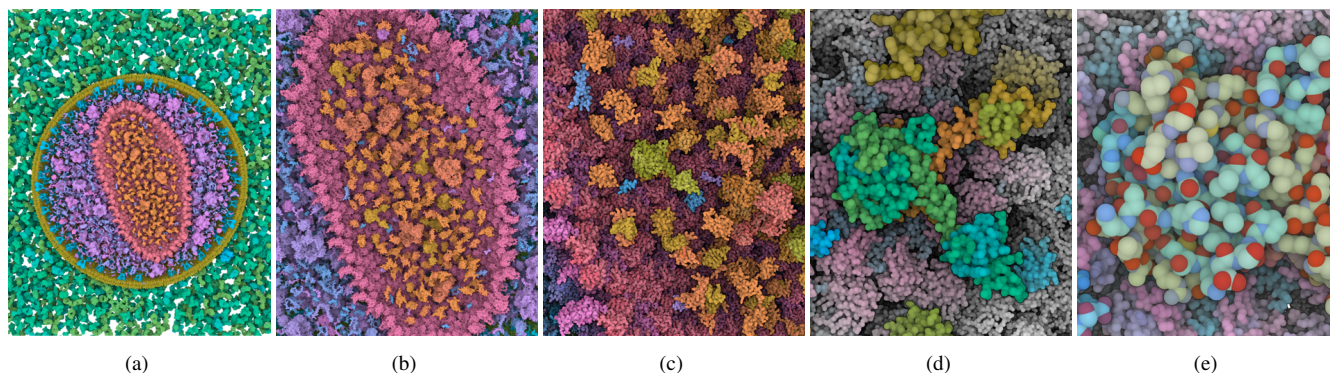


Figure 1: Interactive multi-scale visualization of HIV using “Chameleon” for dynamic color coding of protein and atom properties: Zooming from an overview of the entire virus (a) to the capsid (b), the capsid interior (c), a single integrase protein in the capsid (d), and its atomic structure (e). Mind how protein colors become more differentiated as we zoom in (a-b), and how protein domains (c), secondary structures (d), and individual atoms (e) are being progressively revealed through dynamic color adaptations.

Abstract

Visualization of structural biology data uses color to categorize or separate dense structures into particular semantic units. In multiscale models of viruses or bacteria, there are atoms on the finest level of detail, then amino-acids, secondary structures, macromolecules, up to the compartment level and, in all these levels, elements can be visually distinguished by color. However, currently only single scale coloring schemes are utilized that show information for one particular scale only. We present a novel technology which adaptively, based on the current scale level, adjusts the color scheme to depict or distinguish the currently best visible structural information. We treat the color as a visual resource that is distributed given a particular demand. The changes of the color scheme are seamlessly interpolated between the color scheme from the previous views into a given new one. With such dynamic multi-scale color mapping we ensure that the viewer is able to distinguish structural detail that is shown on any given scale. This technique has been tested by users with an expertise in structural biology and has been overall well received.

Categories and Subject Descriptors (according to ACM CCS): Visualization [Human-centered computing]: Visualization application domains—Scientific visualization

1. Introduction

With improving technology, more and more data across all levels—from atomic to compartment—is being gathered on viruses and cells. Today it is possible to display an ever increasing amount of

this data at once, up to and beyond an entire human immunodeficiency virus (HIV) [LMAPV15], with its complete macromolecular composition. All levels can be present; we can start at the compartment level and zoom in past the proteins, their domains and secondary structures, all the way to the amino acids and atoms (see

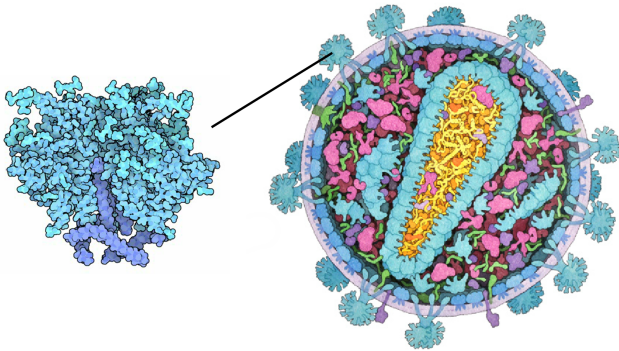


Figure 2: Multi-scale illustration of HIV [RCS11]: overview of the virus on the right and close-up on the envelope protein on the left. In the close-up different shades of blue are used to discriminate between protein domains, and carbon atoms are slightly darker.

Figure 1). Visualizations of HIV are not only used for research purposes, but also to communicate its basic properties to a broader audience, as in Figure 2, which is what we are interested in.

Illustrators carefully select color assignments for each detail level of their illustrations. In Figure 2, the illustrator depicts different protein types by color in the overview on the right. In the close-up on the left, he uses different shades of blue to distinguish between the protein domains, which are not indicated in the overview. In interactively zoomable visualizations, such a static single-scale color assignment will either lead to a loss or excess of information. For example, if we apply categorical color coding to protein groups, then proteins can be easily discriminated, but secondary structures or atoms are not visually depicted (Figure 3, left). On the other hand, when color-coding individual atoms, the visualization is prone to salt-and-pepper noise in the overview levels (Figure 3, right).

A naive approach to multi-scale color mapping would be to independently define colors for each zoom level and then blend them while zooming. However, several issues can occur: First, with independent color assignments for each zoom level, colors may change significantly when zooming. This may not only cause disorientation when zooming due to considerable color changes, but may also cause artificial hues or grayed out colors when blending between levels [CWM09]. Second, there is a loss of hierarchy information when transitioning from one level to the next. For example, when looking at two neighboring compartments and zooming in, it may be hard to maintain an overview of which compartment certain proteins belong to. To take the data hierarchy into account, Tennekes and de Jonge [TdJ14] proposed a color space for hierarchical visualizations. Their approach was targeted towards static visualizations and therefore assigns unique hues to each child node in a tree. However, this leads to poor scalability with respect to color discriminability. The HIV consists of 46 protein types, each divided further into multiple protein domains, secondary structures, amino acids, and atoms. However, this scalability issue can be resolved by exploiting the interactive navigation capabilities – and thereby caused dynamic visibility changes – of our system.

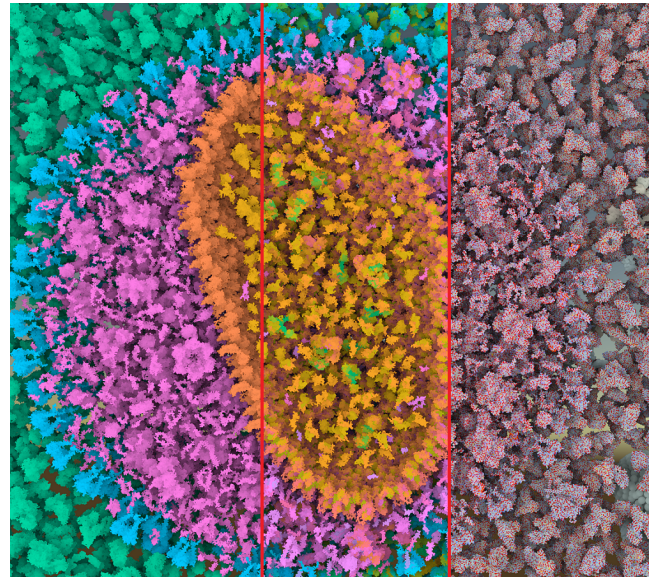


Figure 3: Comparison image of HIV with coloring based on compartments (left), secondary structures (middle), and atoms (right). All different levels can be seen in Figure 10.

We propose the semantic color zooming method Chameleon for multi-scale visualizations of cell or virus-like structures, based on a view-dependent, hierarchical color scheme. Starting from the highest hierarchy level, we progressively free up unused color space and redistribute it to visible elements defined by the current zoom level and item visibility. Our contributions are:

1. A view-dependent, hierarchical hue palette for expressive color mapping of multi-scale structural biology models,
2. Usage of chroma and luminance to minimize visual clutter and encode structural properties across multiple zoom levels, and
3. Results from a small expert user study, showing that users can distinguish structures on multiple scales through dynamic color mapping, while rating the visualization as highly appealing.

2. Related Work

The selection of a color map for data depends heavily on the visualization goal. Not only clear visibility, but also harmonious and appealing colors can be important [Iha03], as well as the assignment and mixing of colors [WGM*08]. There are many points to consider, and as a result guidelines have been published on how to decide which color scheme is most appropriate [BRT95], [Hea96]. There have also been guides aimed at two dimensional color maps, such as by Bernard et al. [BSM*15] and the software ColorMap-Explorer by Steiger et al. [SBT*15]. Furthermore, several tools have been developed to assist users select appropriate colors for maps, such as Colorbrewer [HB03], and color maps for clusters of data, such as “i want hue” [Med16] and XCluSim [LKS*15]. Such clustering can, however, cause perceptual issues, either due to the large number of clusters or their layout. In cases such as colored maps, visually dominant structures may visually suppress small groups or areas [LSS13]. Visualization of information such as

can be found on maps can lead to small areas being next to much larger ones. In this case, it may be difficult to spot the small areas, and lowering the saturation of the larger areas allows the small ones to stand out more. Recognition may also be affected by contrast effects which may need to be dealt with [MSK14]. Particular care needs to be taken when dealing with higher dimensional data [MBS*14], so the distances between the clusters do not become distorted.

While there has been intense research on color mapping in visualization in the past, there have been only very few works related to multiscale visualization. Tree Colors use a tree-based method [TdJ14] that attempts to overcome the above issues in the case of hierarchical structures. The tree method uses HCL space (a cylindrical coordinate version of CIELab with Hue, Chroma (Saturation), and Luminance) and gives each cluster a different hue. The root of the tree has no saturation, i.e. grey, and the hue is divided up into different wedges. A hue wedge is assigned to each branch, with some space cut out between branches, in order for there to be a jump in color from one wedge to another. As the tree spreads out, the saturation of the nodes increases, and each level gets its own smaller wedge. Essentially, each level has its own hue ring with a defined chroma. However, the technique was designed for static visualizations, where all colors are visible at the same time. Therefore, the number of possible colors, for example in a close-up, is severely limited, leading to discriminability problems. In contrast, our method allows for dynamic shrinking and growing of wedges based on visibility. In addition, we also allow for overlapping wedges if a clear structural separation is available. This supports discriminability of items through color and shape.

The development of new modeling techniques for three dimensional molecular structures led to a rapid increase in size of the studied datasets. To keep up with the progress, new visualization methods were specifically developed to address this challenge, such as by Lampe et al. [LVRH07] or Falk et al. [FKE13]. An overview of the state-of-the-art in visualization of biomolecular structures is given by Kozlikova et al. [KKL*15].

To support seamless zooming in molecular visualizations, Parulek et al. [PJR*14] introduced seamless transitions between different levels of coarseness for the representation of the molecular surface based on the distance from the camera. Their goal was not only to reduce the computation cost, but also to utilize the shape as a means to reduce high-level details and noise when observing the structure in its entirety. Le Muzic et al. [LMPSV14] extended this approach to allow for real-time rendering of structures with up to tens of billion atoms. cellVIEW [LMAPV15] renders such large-scale molecular scenes using Unity3D, a popular 3D engine. It showcases large molecular structures generated with cellPACK [JAAA*15], a tool developed to procedurally generate accurate and multi-scale models of entire viruses and cells. While these tools provide the foundation to interactively navigate massively packed molecular landscapes, they currently do not have sufficient visual expressiveness to effectively reveal structural information beyond the protein level. Chameleon is built upon cellVIEW, and is an attempt to improve understanding during exploration of multi-scale molecular scenes generated with cellPACK.

3. Chameleon Overview

Mesoscale datasets feature many relevant properties which are present at different scale levels. The molecular properties that we provide are, from the largest to the smallest scale level: compartment (or protein group), protein type, protein domain, secondary structure, and atom type. According to domain experts, these are the most relevant properties. Scientific illustrators are using these properties to modulate the color palette when generating static visualizations of proteins at different levels, as shown in Figure 2. However, this requires manually setting up the color palette for each level, which can be extremely tedious for an interactively zoomable visualization of an entire virus. Our goal is to provide a dynamic coloring mechanism that automatically visualizes the most relevant information for any given zoom level, thereby fulfilling the following requirements:

1. On each zoom level, associated main structures (i.e., those structures that result in the best visibility in terms of screen size) should be clearly discriminable from each other.
2. When navigating, the transitions between the zoom levels should be smooth to avoid abrupt appearance changes of the visualization or orientation loss.
3. The inherent hierarchy of the structures should be reflected in the color coding.
4. The visualization should be aesthetically pleasing to engage a broad audience.

We manipulate all three color components of the HCL color space to fulfill these requirements: We use the **hue** to distinguish between structures at the respective zoom levels, as described in Section 4. The **chroma** is used to increase color discriminability for elements in the focus against the context (Section 5). Finally, we indicate structures from lower hierarchies in the **luminance** channel to mimic illustrator techniques (see Figure 2) and to smoothen color transitions (Section 6). All objects in the scene must be fully opaque, as transparency for densely packed molecular data can lead to unpredictable and confusing color mixtures. We use CIELab and HCL color palettes, which are perceptually linear. This means we can quantify the dissimilarity between colors by their Euclidean distance. We describe how we dynamically adapt the hues as the user is panning the scene to ensure color discriminability even in a scene with dozens of different entities (Section 4.1), and how hues are sub-divided hierarchically as the user zooms further into the scene (Section 4.2). Finally, we demonstrate the effectiveness of our technique in a user study with professionals and students from the field of molecular biology (Section 8).

4. Dynamic Hue Palette

At the highest hierarchy level (compartment, as illustrated in Figure 1a), our task is to find colors for each individual protein type that allow users to discriminate the proteins, but also understand their hierarchical grouping into the compartments. To visually categorize the protein types, we initially assign each protein type within each compartment a unique hue value. Hues are selected from an iso-luminant hue ring, centered in the gray spot, such as shown in Figure 4. To maintain good contrast for encoding shape properties through shading, we fixed the luminance value to 75. With this luminance value, we also can use a wide radius to fit the hue ring



Figure 4: Iso-luminant hue ring with 21 distinct colors, with luminance 75 and chroma 40 in HCL color space.

within the sRGB gamut, leading to a maximum chroma of 40. Figure 5 shows a comparison between a hue palette picked from an HSL ring (left) and an iso-luminant HCL ring (right). Mind how the structural properties are better visible in the iso-luminant pastel color palette on the right.

For the initial color assignment of protein types associated with individual compartments, we allocate hues from the hue ring to protein types with respect to four parameters: 1) the number of compartments n , 2) the number of protein types m_i within each compartment $1 \leq i \leq n$, 3) the optimal color distance d between structures, which will be elaborated below, and 4) the minimum color spacing s between compartments. The first parameter defines the number of “wedges” that are overlaid on the hue ring, where each wedge represents the color range of a compartment, as illustrated in Figure 6, top row. The second parameter controls the relative angular width of the wedges. The third and fourth parameter control the actual angular width of the wedges: Given the chroma C^* (i.e., radius) of the hue ring and the optimal Euclidean color distance d , the optimal angular distance between neighboring proteins within the same compartment is defined as:

$$\alpha_{opt} = \arcsin \frac{d}{C^*}. \quad (1)$$

In CIELab color space the “just noticeable difference” (jnd) is assumed to be 2.3 [LSS13]. Using the jnd would of course lead to poor discernibility. We discovered that roughly 5 jnd steps were needed to allow for easy discernibility. With this setting and a chroma radius of $C^* = 40$, α_{opt} is 16.7° , which allows for a maximum of 21 distinct hues. However, in the HIV, we have 46 distinct protein types $\sum_i m_i$ distributed to five compartments $n = 5$. We use a minimum spacing $s = 5^\circ$, so that our angular distance α between protein type colors is therefore below the minimum α_{opt} :

$$\alpha = \frac{360 - n \cdot s}{\sum_i m_i}. \quad (2)$$

This means that in the initial overview zoom-level protein colors can be very similar and hard to distinguish. In the overview zoom level, this effect is actually desirable to avoid salt and pepper noise. However, as we zoom in, we want increasing discriminability. We therefore introduce a view-dependent color adaptation approach to increase α dynamically, based on the current visibility.

4.1. View-Dependent Color Adjustment

As the user zooms in and explores the scene, large parts of the scene may not actually be visible in the current viewport, as is illustrated

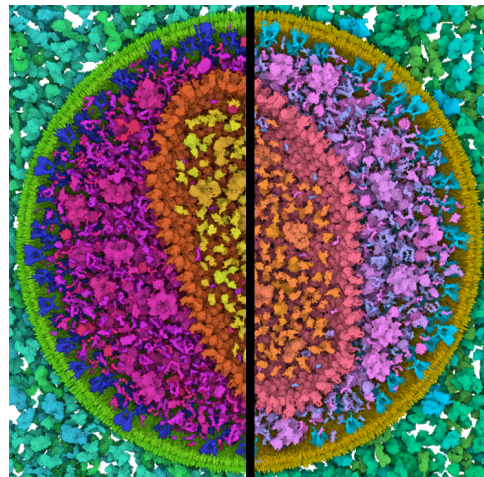


Figure 5: Comparison between two color palettes with varying hues and constant saturation/lightness using HSL (left), and constant chroma/luminance using HCL (right). Unlike HSL, the HCL color space provides true isoluminant colors when varying the hue with constant chroma/luminance values.

in the lower row of Figure 6(a-c). In this configuration the color palette becomes underexploited. We therefore re-compute and optimize the color palette on-the-fly as proteins appear and disappear. An overview of the mechanism is illustrated in Figure 6.

To determine which protein groups occupy unnecessary space in the color palette, we extract visibility information by analyzing the previously rendered frame. For each protein group, we determine the number of visible protein types m_i . A protein type is visible, if at least one screen pixel is occupied by a protein of type i . According to Equation 2, the angular distance α between the hues increases as the number of visible protein types across all protein groups $\sum_i m_i$ decreases.

As visualized in Figure 6d, the hue ring is divided proportionally to the protein groups’ numbers of visible proteins. As the number of visible protein types can be potentially small, we allow the user to determine a maximum angular distance of a protein group to avoid excessive color changes while panning.

4.1.1. Extracting Visibility Information

We leverage GPU computing in a post-processing operation, in order to compute the visibility of each protein type efficiently. Upon rendering, we generate an additional offline texture which contains, for each pixel, the unique identifier of the rendered protein. We primarily declare two GPU buffers which will, for each protein type, store the occupied pixel count and the total number of visible proteins, respectively. Subsequently, in a dedicated compute shader, we iterate through all the pixels, and we increment the pixel counter corresponding to the protein type stored in the video memory. At the same time, we also flag in a dedicated GPU buffer the proteins whose unique IDs are present in the generated texture. This information will allow us to determine the number of visible instances for each protein type. In a second pass and in a dedicated compute

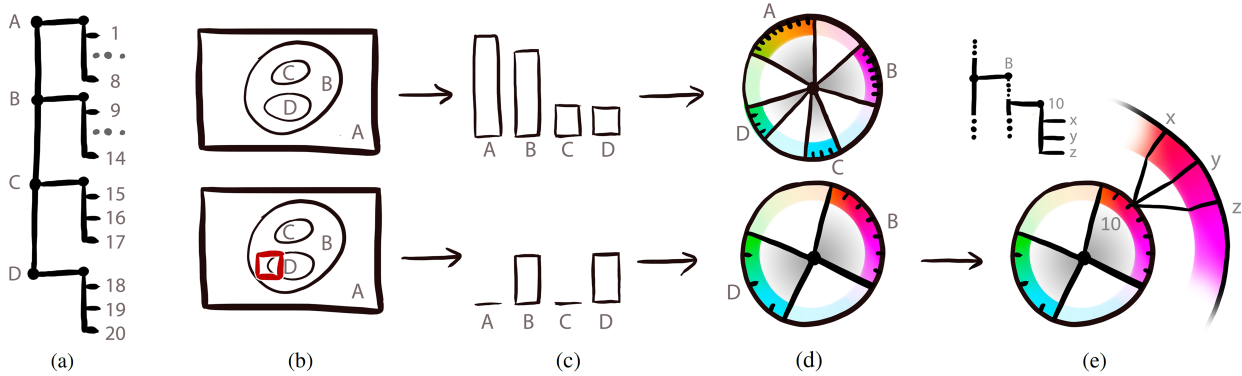


Figure 6: Overview of Chameleon's dynamic and hierarchical hue selection scheme for (a) an exemplary two-level hierarchy of four protein groups (A,B,C,D) and a total number of 20 different protein types: For the current viewport (b), visibility information is extracted (c) and the color palette is updated. Zooming further towards protein type 10 splits up the protein type hue to three different domain colors (x,y,z).

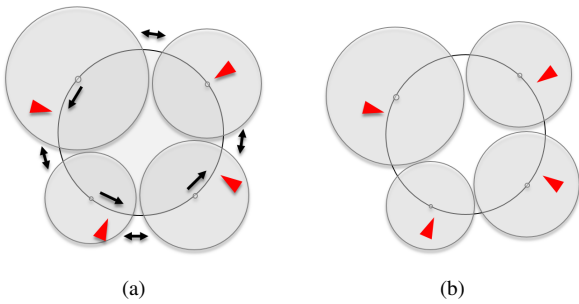


Figure 7: Force-based hue assignment: Initial wedge placement (a) and resulting force-based placement (b). Red arrows indicate the default hues for the wedge centroids, and black arrows represent the forces.

shader we then iterate through all the protein instances. For each protein which was flagged as visible in the previous pass, we increase the number of visible instances corresponding to the protein type. Since the computation is done in parallel it is important to use atomic intrinsic functions to increment the counters, in order to avoid concurrent access to the same value from two different threads.

4.1.2. Force-Based Hue Assignment

As we move the camera and affect the visibility of the proteins, the angles of the protein-group wedges on the hue ring are dynamically growing and shrinking. To determine the optimal location of the protein-group wedges on the hue ring, we use a force-based method to reposition the wedge centroids on the ring. We start by positioning a first wedge on the circle, and then place the remaining wedges subsequently on the circle, as illustrated in Figure 7a. While the position of the first wedge on the hue ring can be arbitrarily defined with this approach, the other wedges are not guaranteed to be located as near as possible to their default centroid (i.e., the initially assigned wedge centroid with all visible protein types).

This could lead to unnecessarily large hue shifts for some protein groups while navigating, as shown in Figure 7a.

To avoid this issue, we use a force-based layout for the positioning of the group centroids on the hue ring. The forces are applied in a single dimension, i.e., along the hue ring, and include distance constraints between the groups as well as attraction forces between groups and their default positions. We then integrate the forces, using Euler Integration to compute the new positions of the groups until reaching a state of equilibrium. For a given group i , the distance-constraint response-force with another group j is computed as follows:

$$y(i, j) = l(g_i, g_j) \max((r_i + r_j) - |g_i - g_j|, 0), \quad (3)$$

where l corresponds to the direction of the response, i.e., whether the hue angle should increase or decrease, g is the position of the group centroid, and r is the radius of the distance constraint and corresponds to the protein group's wedge radius plus the spacing between group wedges. In other words, the force-based hue assignments aims to keep the resized wedges close to their default centroids, while avoiding overlaps between neighboring wedges. For a single group, the overall forces – comprising of attraction and repulsion forces – are computed as follows:

$$f_i = (g_i - I_i) + \sum_{j=1}^{n, i \neq j} y(i, j) \quad (4)$$

The first part of the right side of this equation corresponds to the attraction forces, where I corresponds to the initial group position. The second part is the sum of repulsion forces between the given group and the other ones. Additionally we also include a damping value of 0.2 (reduce velocity by 20% per time step) in the integration calculus to smooth-out the motion of the group centroids when the viewport and visibility configuration abruptly change. This does not affect the final color assignment for a view, merely the transition.

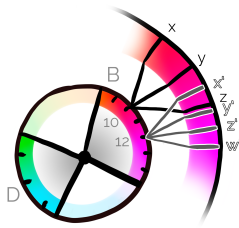


Figure 8: Hierarchical color assignment for the protein chains. The spacing between the protein domains (x,y,z) and (x',y',z',w') is predetermined based on the number of domains, and domain-hue wedges can also overlap.

4.2. Hierarchical Colors

As we zoom further into the details of the proteins, we begin to reveal the colors corresponding to the following levels (domains, secondary structures, atoms). To ensure logical transitions between levels and to maintain hierarchical information, we introduce a hierarchical hue picking method. For each protein type, the hue ring is further sub-divided into wedges to color-code protein domains, and individual protein-domain hues can further be sub-divided to secondary-structure wedges.

To select hues for m protein domains, the angular distances γ between the domain hues are calculated as follows:

$$\gamma = \min(\alpha_{opt}, \beta/m)$$

where β is the maximum size of the domain wedge and α_{opt} is the optimal angular distance, as defined in Section 4. This means that the wedge will grow linearly with the number of domains until it reaches the angle β , at which point the angle between domains will start to shrink. We set β to 180 so that a single protein would not cover more than half of the hues. A larger β could lead to a loss of hierarchical information. Choosing a lower β may cause domains within a single protein to be hardly distinguishable. The domain hue wedge is centered around the protein hue value, as illustrated in Figure 8. Mind that domain hue wedges can overlap across proteins. This is another major difference to the hierarchical coloring technique introduced by Tenekes and de Jonge [Tdj14], who assigned unique, yet hardly distinguishable colors on all zoom levels for static tree visualizations. In our case, being able to clearly distinguish domains is an important task in structural biology, which is difficult without clearly distinguishable colors. To overcome mis-interpretation caused by overlapping hue wedges between proteins, we added context-de-saturation on demand.

4.2.1. Interpolating Between Levels

The individual zoom levels (compartment, proteins, domains, etc.) are associated with specific distances of the camera to the scene. In our HIV example, distance values were chosen so that the structures associated with the zoom levels were optimally visible. Assigning zoom levels to camera distances is task-dependent. Some items, like atoms and amino acids, have very similar sizes (see Figure 9) and would therefore occupy similar zoom ranges, i.e. it is not possible to uniquely determine what information a user would be

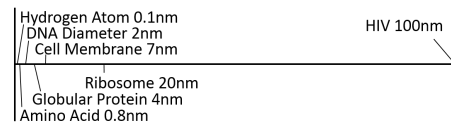


Figure 9: Illustration of the scales of HIV. Note that some items have very similar sizes, like atoms and amino acids.

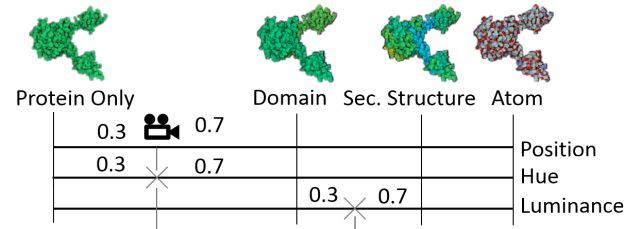


Figure 10: Interpolation of hue and luminance values for a fragment: based on the world space distance to the camera, the interpolation factor between the two closest levels (protein and domain) is computed and used as blending factor between protein and domain hue, as well as domain and secondary structure luminance offset.

interested in for some camera distances. Thus, it depends on the users' priorities to select or omit certain levels for color coding.

To smoothen the transition between zoom levels, we interpolate the hue values in between the levels. This transition is performed in post-processing on the GPU, in a dedicated shader program. In addition to the unique protein identifier texture mentioned in Section 4.1.1, we also generate an additional offline texture upon rendering, which contains the unique identifier of each rendered atom. While iterating through all the pixels, we fetch their unique atom identifiers from the offline texture. The identifiers allow us to collect, directly from the shader program, all the needed information concerning the rendered atoms, and the identifiers were priorly uploaded to the video memory (group, protein type, domain, secondary structure, and atom type). We also use the previously rendered depth texture to retrieve the world-space position of a pixel in the shader. Then, from the world position of each pixel, we compute the world space distance to the camera. From the camera distance we are able to compute the current degree of transition between the closest levels, as well as the hue values associated with the two closest zoom levels. Finally, the hue of a fragment is computed by performing a linear interpolation of the hue value of the two levels, as illustrated in Figure 10.

In our current implementation, users have to define the camera distances for the zoom levels manually. In the future, we plan to investigate automatic camera-distance settings, based on the screen size of the structures associated with the zoom levels.

4.2.2. Pre-Defined Color Palettes

In some cases, it can be desirable to use pre-defined color palettes instead of automatically calculated categorical colors. In our HIV showcase, we use pre-defined atom color codes for the highest zoom level (see Figure 1e), to support domain knowledge, such as

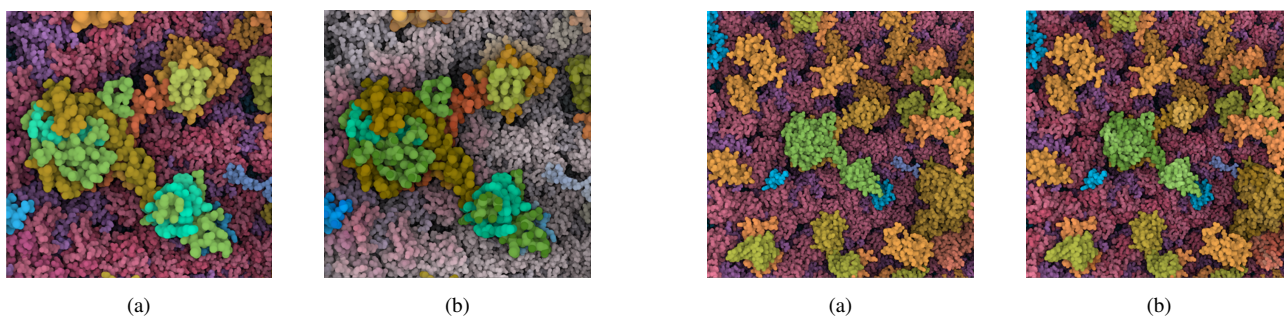


Figure 11: Without (a) and with (b) desaturated peripheral colors.

CPK coloring for atoms. In this case, Chameleon simply blends to those pre-defined colors when zooming to the atomic level.

5. Context Desaturation

Due to our hierarchical color assignment, domain colors of neighboring proteins or secondary structures of neighboring domains may be assigned identical colors (Figure 8). To increase color discriminability, and reduce occurrence of similar colors, we modulate the chroma to generate a focus+context effect. By clicking on a protein of interest, we define a spherical 3D region around the protein, which constitutes the focus region. Outside this region, we progressively decrease the chroma value down to 0 as we zoom towards the focused protein, to ensure optimal color discriminability of the focus protein's colors with respect to all other colors in the scene, as can be seen in Figure 11.

6. Luminance Modulation

So far, we used the hue channel to discriminate structures on different zoom levels and the chroma to distinguish focus proteins from the context. The final color channel we manipulate is luminance. Molecular illustrators sometimes use the luminance channel to indicate structural properties in lower hierarchy levels, as shown in the close up in Figure 2 or in the "Molecule of the Month" [Goo13]. This is a reasonable choice, given that hue is known to be an effective channel to encode low-frequency information, while luminance is more effective to encode high-frequency information [BRT95]. In addition, encoding information from an additional zoom level can support orientation when blending between zoom levels.

For each hierarchy level, we therefore not only assign unique hues, but also unique luminance values. We modulate our base luminance value of 75 by up to a value of 15, as illustrated in Figure 12. With this small modulation, the luminance channel serves only as a subtle indication of lower-level structural properties, to avoid extensive visual clutter and not to interfere with shading cues.

Like the hue, the luminance value is defined by the distance of the camera to the respective structures. Given an interpolation factor derived from the zooming distance to the pre-defined zoom levels (see Figure 10), luminance values are interpolated between the next two hierarchy levels with the current interpolation factor, as

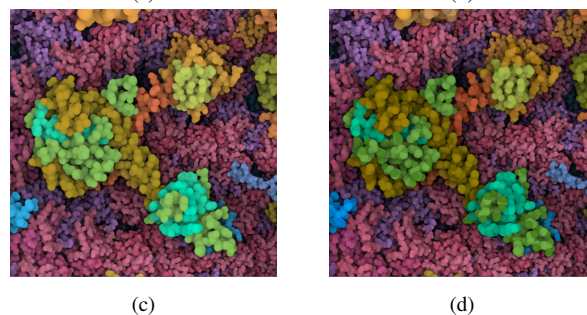


Figure 12: Comparison of domain level (top) and secondary structures level (bottom) without (left column) and with (right column) luminance modulation.

illustrated in Figure 10, bottom line. This way, the hue and the luminance encode different hierarchy levels and thereby generate an effect similar as shown in the close up in Figure 2.

7. Results

To showcase the capabilities of the Chameleon, we applied it to a mesoscale molecular scene. The dataset was provided by domain experts and modeled with cellPACK [JAAA*15], a tool for the procedural generation of large biomolecular structures, incorporating the most recent knowledge from structural and systems biology. We use cellVIEW [LMAPV15] for the rendering part, which is a tool designed to efficiently render large molecular scenes on the GPU and is implemented in Unity3D.

The dataset which we showcase is a model of an HIV particle in the blood serum which contains 20502 instances of 46 different protein types including two lipid membrane layers. The protein types are grouped together based on their location. There are 6 different protein groups in the dataset; the blood plasma proteins (18 types), the lipids (2 types), the envelope surface (5 types) and interior proteins (15 types), the capsid surface (1 type) and proteins inside the capsid (5 types). Figure 1 shows progressive zooming from the entire virus to single atoms. Initially, the blood plasma proteins in green, and the matrix protein in purple, dominate the color palette because they have more protein types than other compartments, like the orange capsid proteins. As we get closer to the capsid proteins, we observe that the colors become more distinguishable. To visualize the results of the view-dependent color-palette optimization more precisely, we also developed a palette widget, shown in Figure 13, bottom left.

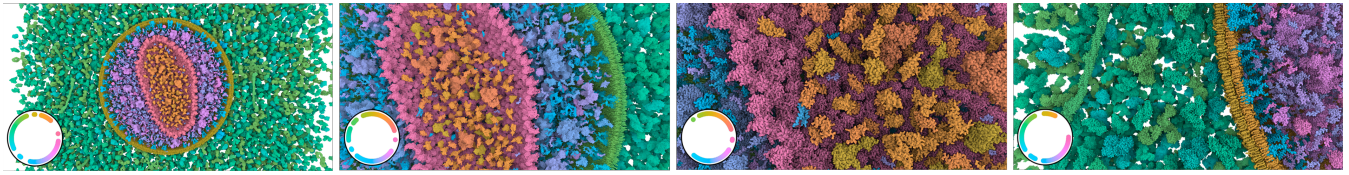


Figure 13: Various snapshots from the Chameleon system. As the camera position changes, the discriminability between protein types is optimized. The color palette widget at the bottom left allows us to visualize the distribution of the groups along the hue ring. The hue ring is not intended to be directly interpreted. It is a technical complement to support intuitive understanding.

Chameleon relies on GPU computing to provide a smooth and responsive user experience. Since we perform all the coloring operations in post-processing, the size of the dataset has only little impact on the performance. We benchmarked the computation speed at HD resolution (1920×1080) on a machine equipped with an Intel Core i7-3930 CPU 3.20 GHz machine coupled with a GeForce GTX Titan X graphics card with 12 GB of video RAM. We measured **0.5 ms** on average to count the visible instances when zooming out on the entire dataset, which is the most extreme case since a very large number of proteins are visible. It took **1.2 ms** to compute the screen pixel coverage of the protein types with the same configuration. The last stage of the pipeline, in which we assign the final hue, chroma, and saturation values to each pixel, took **1.3 ms** on average. This also includes the cost of reading back the visibility information to the CPU for the force-based hue assignment.

8. User Study

Since interactively explorable multi-scale visualizations of biology models have only become available very recently, there is no comparable approach how to represent biological structures across multiple scales. Dynamic visual discrimination, apart from geometric levels of detail [PJR*14], have not been studied so far in the biological field. We therefore decided for a qualitative evaluation since there is no clear baseline for a comparative lab experiment, in order to answer two research questions:

1. Does Chameleon color mapping support discrimination of structures on multiple scales?
2. Are the dynamic color changes by Chameleon distracting / unpleasant in the exploration process?

To answer these two research questions, we asked five students and professionals in the field of biology to perform two tasks. In the first task (*structure identification task*), users were asked to identify structures on multiple levels of detail. We based our task description on the a previous publicly available description of the HIV capsid at PDB-101 [Goo13] Based on this description of the HIV capsid, users were asked to identify the following structures:

1. one of the 12 pentamer capsid proteins,
2. the N-terminal and C-terminal domain of the capsid protein,
3. the alpha helix in the N-terminal domain stabilizing the hexamers / pentamers,
4. the binding site of Cyclophilin A, which is a loop on the surface of the capsid protein with several proline amino acids, and
5. one methionine amino acid within the alpha helix.

The last item was not included in the expert's task description, but was added as a representative task on the amino acid and atom level, respectively. Since we did not color-code the amino acid level explicitly, users were given the hint to identify methionine based on the coloring of its sulfur atom. To assess the performance in the structure identification task, we recorded whether users were able to correctly identify the above listed structures. In the subsequent *free exploration task*, users could freely navigate through the visualization, while thinking aloud. All reported insights were noted. Both tasks were video-taped and followed by a questionnaire and a semi-structured interview. Before the study, users could play around with the tool to get familiar with the navigation, and were also instructed how to toggle the visibility of the protein groups.

With the structure identification task, we could assess whether experts with sufficient knowledge to understand molecular structures without additional text labels are able to identify the above structures using our system. While our color mapping provided the necessary discrimination of individual structures in the respective zoom levels, the identification of the structures was only possible through their structural properties. Through the free exploration task, our goal was to assess whether users would notice and be distracted by our dynamic color mapping.

Mind that structural information alone is not sufficient to identify structures below the protein level, as shown in Figure 14. In addition, performance measures of such complex tasks are rather hard to compare with only a small number of expert users performing the study.

We report observations, questionnaire results, and feedback from five experts or students in the field of molecular biology (2 PhD students, one Post-Doc, one pharmacist, and one master student; one female, aged 24 to 31, all with normal or corrected-to-normal vision). While the method is also aimed at the general audience, no such users were involved in the study. The task description for such users would have to be rather long and detailed, leading likely to issues with text comprehension or revealing too much information.

8.1. Results

Table 1 summarizes the performance of users in the structure identification task. Except for user 3 (the pharmacist), all users were able to correctly identify all structures down to the secondary structure level. User 3 mixed up the N and C terminals of the capsid protein, and the rest of the tasks are to some degree based on finding them. With regards to the amino acid, there was more than one kind

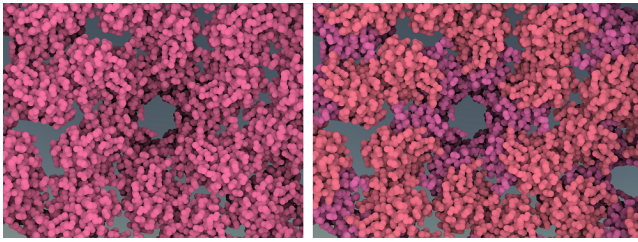


Figure 14: Comparison between static protein coloring (left) and hierarchical color coding using Chameleon (right) for viewing a capsid protein hexamer on the protein domain level.

Table 1: Structure identification task: **c** identified correct item, **p** partially correctly identified item, **i** identified incorrect item, **n** nothing found. User-reported certainty shown on a scale of 1 (lowest) to 5 (highest).

Task \ User	1	2	3	4	5
Pentamer	c (5)	c (5)	i (4)	c (5)	c (5)
C-terminal	c (5)	c (4)	i (1)	c (4)	c (4)
N-terminal	c (5)	c (4)	i (1)	c (2)	c (4)
α -helix	c (1)	c (3)	n	c (4)	c (2)
Binding site	c (1)	c (5)	c (5)	c (5)	c (4)
Methionine	c (1)	p (5)	n	p (1)	n

of amino acid with a sulfur atom, and two users mistook it for the correct one.

In the questionnaire, users assessed the identification of compartments, but also the proteins, as very easy (see Figure 15). However, the domain and secondary structure identification was rated as much more difficult. This is also reflected in the users' self-reported certainty of the identified structures (Table 1). While the users were quite certain about the identity of a pentamer capsid, once they spotted it, they were least certain about the alpha helices stabilizing the hexamers and pentamers. All users, except for user 3, were able to identify the C- and N-terminal domains of the capsid proteins. All those users verbally referred to the domains by color. Two users also referred to the alpha helices by color.

In the free exploration task, all except for user 3 explored the virus. On average, they spent 10 minutes for exploration. Some users reported that they learned something new when exploring the visualization, such as that "HIV uses [the] host membrane" and "how crowded everything is". The four users focused on different parts of the virus during their free exploration, such as the membrane (user 1 and 5), the proteins in the matrix (user 1 and 4), and the amino acids of the Cyclophilin A binding site (user 2).

In the questionnaire, most users indicated that they the color change (see Figure 15), but the confusion caused by these color changes were rated fairly low. When asked about the changing colors in the post-experiment interview, all users reported that they noticed changing colors when zooming to the atomic level. However, only one user noticed it on all levels, while a second one thought "something odd" was occurring on the secondary structure level.

In general, our users all issued the highest possible grade for visual appeal in our questionnaire. In the post-experiment interview, they explained that they considered the tool useful for presenting

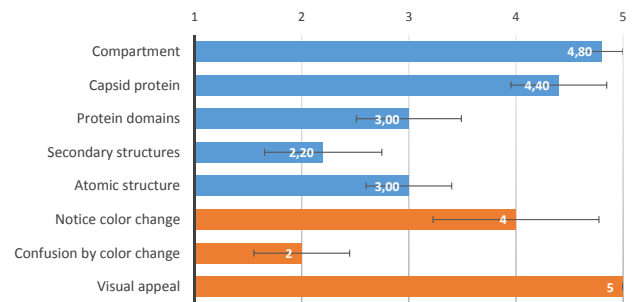


Figure 15: Average and standard error of responses on five-point Likert scale for questions concerning the ease of identification of structures (blue), as well as color changes and visual appeal (orange).

their research and educating students – which is in line with our research goals. However, they also had suggestions for improvement, such as adding text labels, visually marking the termini of the protein domains, and providing a cartoon representation for secondary structures.

8.2. Summary and Discussion of Results

The performance and feedback of the molecular biologists in our study indicate that Chameleon supports users in identifying molecular structures on multiple scales. Users were equally successful identifying one of the capsid proteins forming a pentamer, and identifying the two domains of the capsid protein. The pentamer capsid protein differed only in shape from the more frequent hexamer capsid proteins, while the two different domains of the capsid proteins were encoded by color. However, the users reported lower certainty and higher task difficulty for the identification of the domains. This is an indication that identification by structure is easier than identification by color alone – yet, color can be used if no strong structural cues are available, as in our protein domain example.

On the secondary structures level, the identification rate was similarly high, but users were quite uncertain about their findings and reported a high task difficulty. On this level, alpha helices and beta sheets were discriminated by color, similar to Figure 12. However, the particular alpha helix mentioned in the task description could only be identified as a helix by shape. The request for an alternative cartoon representation for identifying structures on this level shows the limits of multi-scale color mapping without adapting the structural representation. In the future, it will be important to explore combinations of semantic zooming comprising both, structural and color information.

User feedback shows that our system's dynamic color changes do not interfere with the users' workflow. While protein domains and secondary structures were mainly referred to by their color, only few users noticed the color changes (except at the atomic level) and none found them confusing. All users found the visualization highly appealing and useful for presentation and education purposes. We therefore conclude that Chameleon is a valuable extension to multi-scale molecular visualization and does not cause

any notable distraction, but allows for efficient and pleasing visual encoding of protein sub-structures.

9. Conclusion and Future Work

The work presented here is a method that is capable of visualizing a hierarchical, cell-like structure on different levels in a coherent and cohesive manner using color. The hierarchical structure of the coloring technique, along with the visibility based subtle variation in color allows users to navigate and inspect parts of the visualization without getting disorientated, while being subtle. Altering the color based on semantic zooming changes the meaning of color coding. Instead of discriminability for any situation, semantic color changes allow structures to be clustered or separated as necessary. This is achieved while maintaining the connection between structures on different levels in a logical manner via the hierarchical nature of the coloring. Furthermore the coloring scheme is capable of showing structures on each level in a visually distinct manner. The user study performed with our research prototype shows that experts can find information in an HIV dataset on each level, while not being distracted by the dynamic color mapping.

While we demonstrated Chameleon on our HIV showcase with five zoom levels, we plan to extend our approach to more general use cases in the future – in the biology domain and beyond. Currently, the hues of the overview level are initially assigned using a force-based layout and are sub-divided at lower hierarchy levels. In the future, we plan to investigate methods with consistent hue assignments across all levels.

Acknowledgments

The first two authors contributed equally. This project has been funded by the Vienna Science and Technology Fund (WWTF) through project VRG11-010 and supported by EC Marie Curie Career Integration Grant through project PCIG13-GA-2013-618680 and the Austrian Science Fund FWF through project T 752-N30.

References

- [BRT95] BERGMAN L. D., ROGOWITZ B. E., TREINISH L. A.: A rule-based tool for assisting colormap selection. In *Proceedings of the 6th conference on Visualization '95* (1995), IEEE Computer Society, p. 118. 2, 7
- [BSM*15] BERNARD J., STEIGER M., MITTELSTÄDT S., THUM S., KEIM D., KOHLHAMMER J.: A survey and task-based quality assessment of static 2d colormaps. In *IS&T/SPIE Electronic Imaging* (2015), International Society for Optics and Photonics, pp. 93970M–93970M. 2
- [CWM09] CHUANG J., WEISKOPF D., MÖLLER T.: Hue-preserving color blending. *Visualization and Computer Graphics, IEEE Transactions on* 15, 6 (2009), 1275–1282. 2
- [FKE13] FALK M., KRONE M., ERTL T.: Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 195–206. 3
- [Goo13] GOODSSELL D.: Pdb-101 molecule of the month: Hiv capsid. <http://pdb101.rcsb.org/motm/163> (2013). [Online]. 7, 8
- [HB03] HARROWER M., BREWER C. A.: Colorbrewer.org: an online tool for selecting colour schemes for maps. *The Cartographic Journal* 40, 1 (2003), 27–37. 2
- [Hea96] HEALEY C. G.: Choosing effective colours for data visualization. In *Visualization '96. Proceedings.* (1996), IEEE, pp. 263–270. 2
- [Iha03] IHAKA R.: Colour for presentation graphics. In *Proceedings of DSC* (2003), p. 2. 2
- [JAAA*15] JOHNSON G. T., AUTIN L., AL-ALUSI M., GOODSSELL D. S., SANNER M. F., OLSON A. J.: cellpack: a virtual mesoscope to model and visualize structural systems biology. *Nature methods* 12, 1 (2015), 85–91. 3, 7
- [KKL*15] KOZLIKOVA B., KRONE M., LINDOW N., FALK M., BAAEDEN M., BAUM D., VIOLA I., PARULEK J., HEGE H.-C., ET AL.: Visualization of biomolecular structures: state of the art. In *Eurographics Conference on Visualization (EuroVis)-STARs* (2015), The Eurographics Association. 3
- [LKS*15] L'YI S., KO B., SHIN D., CHO Y.-J., LEE J., KIM B., SEO J.: Xclusim: a visual analytics tool for interactively comparing multiple clustering results of bioinformatics data. *BMC bioinformatics* 16, Suppl 11 (2015), S5. 2
- [LMAPV15] LE MUZIC M., AUTIN L., PARULEK J., VIOLA I.: cel-view: A tool for illustrative and multi-scale rendering of large biomolecular datasets. In *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine* (2015), VCBM '15, Eurographics Association, pp. 61–70. 1, 3, 7
- [LMPSV14] LE MUZIC M., PARULEK J., STAVRUM A.-K., VIOLA I.: Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 141–150. 3
- [LSS13] LEE S., SIPS M., SEIDEL H.-P.: Perceptually driven visibility optimization for categorical data visualization. *Visualization and Computer Graphics, IEEE Transactions on* 19, 10 (2013), 1746–1757. 2, 4
- [LVRH07] LAMPE O. D., VIOLA I., REUTER N., HAUSER H.: Two-level approach to efficient visualization of protein dynamics. *Visualization and Computer Graphics, IEEE Transactions on* 13, 6 (2007), 1616–1623. 3
- [MBS*14] MITTELSTÄDT S., BERNARD J., SCHRECK T., STEIGER M., KOHLHAMMER J., KEIM D. A.: Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis. In *EuroVis - Short Papers* (2014), The Eurographics Association. 3
- [Med16] MEDIALAB: i want hue. <http://tools.medialab.sciences-po.fr/iwanthue/> (2016). [Online]. 2
- [MSK14] MITTELSTÄDT S., STOFFEL A., KEIM D. A.: Methods for compensating contrast effects in information visualization. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 231–240. 3
- [PJR*14] PARULEK J., JÖNSSON D., ROPINSKI T., BRUCKNER S., YNNERMAN A., VIOLA I.: Continuous levels-of-detail and visual abstraction for seamless molecular visualization. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 276–287. 3, 8
- [RCS11] RCSB PDB: The structural biology of hiv. <http://pdb101.rcsb.org/learn/resource/the-structural-biology-of-hiv-flash> (2011). [Online]. 2
- [SBT*15] STEIGER M., BERNARD J., THUM S., MITTELSTÄDT S., HUTTER M., KEIM D. A., KOHLHAMMER J.: Explorative analysis of 2d color maps. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2015)* (2015), pp. 151–160. 2
- [TdJ14] TENNEKES M., DE JONGE E.: Tree colors: Color schemes for tree-structured data. *Visualization and Computer Graphics, IEEE Transactions on* 20, 12 (2014), 2072–2081. 2, 3, 6
- [WGM*08] WANG L., GIESEN J., McDONNELL K. T., ZOLLIKER P., MUELLER K.: Color design for illustrative visualization. *Visualization and Computer Graphics, IEEE Transactions on* 14, 6 (2008), 1739–1754. 2

PAPER



Cuttlefish: Color Mapping for Dynamic Multi-Scale Visualizations

Cuttlefish: Color Mapping for Dynamic Multi-Scale Visualizations

N. Waldin¹, M. Waldner¹, M. Le Muzic¹, E. Gröller^{1,2}, D. Goodsell³, A. Ludovic³, A. Olson³, and I. Viola¹

¹TU Wien

²VRVis

³Scripps Research Institute

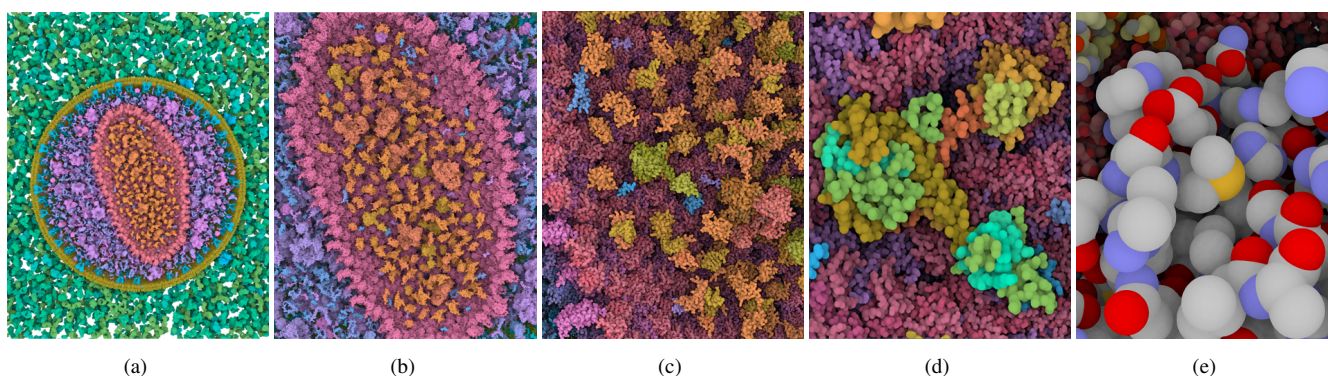


Figure 1: Interactive multi-scale visualization of HIV using dynamic categorical color coding for proteins and atoms: Zooming from an overview of the entire virus (a) to the capsid (b), the capsid interior (c), a single integrase protein in the capsid (d), and its atomic structure (e). Protein colors become more differentiated as we zoom in (a-b), and protein domains (c), secondary structures (d), and individual atoms (e) are being progressively revealed through dynamic color adaptations.

Abstract

Visualizations of hierarchical data can often be explored interactively. For example, in geographic visualization, there are continents which can be subdivided into countries, states, counties and cities. Similarly, in models of viruses or bacteria at the highest level are the compartments, and below that are macromolecules, secondary structures (such as α -helices), amino-acids, and on the finest level atoms. The identification of these items can be assisted through the use of color at all levels. However, currently there are no hierarchical color mapping techniques for very large multi-scale visualizations that can be explored interactively. We present a novel technique for adaptively adjusting the color scheme to the current view and scale. Color is treated as a resource and is redistributed given the current requirements. With such dynamic multi-scale color mapping, we ensure that the user is able to distinguish items on any level. This technique has been tested by users with an expertise in structural biology on an HIV dataset and was overall well received.

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

1. Introduction

With improving technology, data sets are becoming larger and more complex. Furthermore, they are often available in multiple levels of detail. For example, it is possible to visualize an entire human immunodeficiency virus (HIV) [LMAPV15], from the compartment to the atomic level, with its complete macromolecular composition. All levels can be present; we can start at the compartment

level and zoom in past the proteins, their domains and secondary structures, all the way to the amino acids and atoms (see Figure 1). Similarly, geographic information is often visualized on multiple scales, such as quantitative data that can be associated with different levels of detail (e.g., from the continent level down to the district level) [STH03]. Another example for dynamic multi-scale

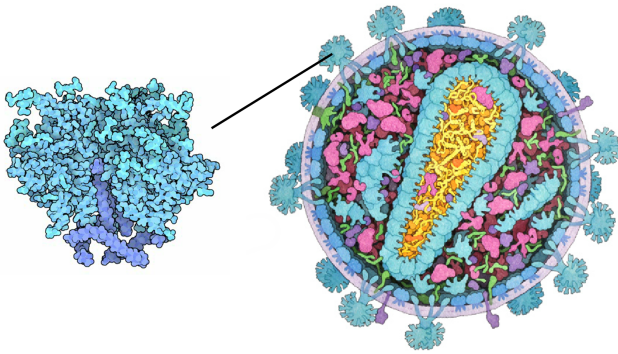


Figure 2: Multi-scale illustration of HIV [RCS11]: overview of the virus on the right and close-up on the envelope protein on the left. In the close-up different shades of blue are used to discriminate between protein domains, and carbon atoms are slightly darker.

visualizations are zoomable treemaps [BL07] and other types of hierarchically aggregated visualizations [EF10].

As such, the color mapping of the data needs to take these different levels into account. Currently, illustrators carefully select color assignments for each detail level of their illustrations. In Figure 2, an illustrator depicts different protein types by color in the overview on the right. In the close-up on the left, he uses different shades of blue to distinguish between the protein domains, which are not indicated in the overview. In interactively zoomable visualizations, such a static, single scale color assignment will either lead to a loss or excess of information. If the coloring depends on the details, the result will be prone to salt-and-pepper noise (Figure 3), while a high level coloring will remove any detail information (Figure 3).

A naive approach to multi-scale color mapping independently defines colors for each zoom level and then blends them while zooming. However, several issues can occur: First, with independent color assignments for each zoom level, colors may change significantly when zooming. This can cause disorientation when zooming due to considerable color changes, and may also cause artificial hues or greyed out colors when blending between levels [CWM09]. Second, there is a loss of hierarchical information when transitioning from one level to the next. For example, when looking at two neighboring compartments in Figure 1a and zooming in, it may be hard to maintain an overview of which compartment certain proteins belong to if their colors are assigned independently of their containing compartments. To take the data hierarchy into account, color spaces for hierarchical visualizations have been proposed in the past [FWR99, TdJ14]. These approaches are targeted towards static visualizations and therefore assign unique hues to each child node in a tree. This leads to poor scalability with respect to color discriminability. In hierarchical data structures the number of items per level grows exponentially. This scalability issue can be resolved by exploiting the interactive navigation capabilities – and thereby caused dynamic visibility changes – of dynamic multi-scale visualizations.

We propose a semantic color zooming method for multi-scale visualizations, based on a view-dependent, hierarchical color scheme.

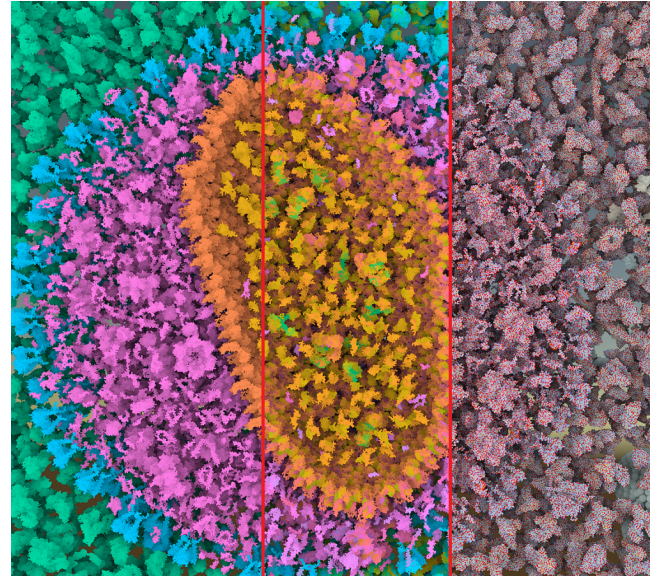


Figure 3: Comparison image of HIV with coloring based on compartments (left), secondary structures (middle), and atoms (right).

Starting from the highest hierarchy level, we progressively free up unused color space and redistribute it to visible items defined by the current zoom level and item visibility. Our contributions are:

1. A categorical view-dependent, hierarchical hue assignment for expressive color mapping of multi-scale visualizations.
2. Two methods how to use the luminance and chroma channels to enhance this color mapping by encoding additional quantitative information or structural properties across multiple zoom levels.
3. Results from a small expert user study, showing that participants can distinguish biological structures on multiple scales through dynamic multi-scale color mapping, while rating the visualization as highly appealing.

2. Related Work

The selection of a color map for data depends heavily on the visualization goal. Clear visibility, and also harmonious and appealing colors are important [Iha03], as well as the assignment and mixing of colors [WGM*08]. There are many issues to consider, and as a result guidelines have been published on how to decide which color scheme is most appropriate [BRT95, Hea96]. There have also been guides aimed at two dimensional color maps, such as by Bernard et al. [BSM*15] and the software ColorMap-Explorer by Steiger et al. [SBT*15]. Furthermore, several tools have been developed to assist users to select appropriate colors for maps, such as Colorbrewer [HB03], and color maps for clusters of non-hierarchical data, such as “i want hue” [Med16]. A simplistic clustering method can, however, cause perceptual issues, either due to the large number of clusters or their layout. In cases such as colored maps, visually dominant items may visually suppress small groups or areas [LSS13]. Visualization of information as given for maps can lead to small areas being next to much larger ones. In this case, it may be difficult to spot the small regions. Lowering the saturation

of the larger areas allows the small ones to stand out more clearly. Recognition may also be affected by contrast effects which may need to be dealt with [MSK14]. Particular care needs to be taken when dealing with higher dimensional data [MBS*14], so the distances between the clusters do not become distorted.

While there has been intense research on color mapping in visualization in the past, there are only few color mapping techniques for multi-scale visualizations. Fua et al. [FWR99], for instance, introduce proximity-based coloring for hierarchical parallel coordinates. Polyines belonging to the same cluster are assigned similar hues, while unused “hue buffers” between clusters ensure that individual clusters can be clearly distinguished. Tree Colors [Tdj14] uses a tree-based method to assign hues to hierarchical data structures. The approach uses the HCL space (a cylindrical coordinate version of the CIELab space with Hue, Chroma, and Luminance) and assigns to each cluster a different hue. The root of the tree is depicted without saturation in grey, and the hue is divided into different wedges. A hue wedge is assigned to each tree branch, with some unused buffer space between branches. As the tree spreads out, the saturation of the nodes increases, and each level gets its own smaller wedge. Essentially, each level has its own hue ring with a defined chroma. These techniques were designed for static visualizations, where all colors are visible at the same time. Therefore, the number of possible colors is severely limited, leading to discriminability problems for very large data sets. In contrast, our method dynamically reassigns colors based on the current visibility. While this increases discriminability of items as the number of visible items decreases, we also take care that the introduced color shifts of individual items are minimized.

Prior work with dynamic color mapping for multi-scale molecular has been proposed by Waldin et al. [WLW*16]. However, it is applicable only to biomolecular models. In this work, we extend the dynamic color mapping approach beyond the biology domain. In addition, we demonstrate how multi-scale categorical color mapping can be combined with a hierarchical quantitative color scheme to convey hierarchically aggregated quantitative information. We demonstrate this for molecular visualization and geographic visualization.

3. Overview

Our goal is to provide a dynamic color mapping technique that automatically visualizes the most relevant information for each zoom level, thereby fulfilling the following requirements:

1. On each zoom level, associated main items (i.e., those items that are best visible at the current level) should be clearly discriminable from each other.
2. The inherent hierarchy of the data structure should be reflected in the color coding.
3. In continuously zoomable multi-scale visualizations, the color transitions between the zoom levels should be smooth to avoid abrupt appearance changes or orientation loss.
4. The coloring must be coherent, i.e., a small shift in the view should not completely alter the colors.
5. The coloring must be consistent, i.e., the color mapping from a specific view should always be the same.

6. The visualization should be aesthetically pleasing to engage a broad audience.

We manipulate the HCL color space to create a hierarchical coloring scheme. HCL (hue, chroma, luminance) is a polar coordinate version of the perceptually uniform CIELab space. This means we can quantify the dissimilarity between colors by the Euclidean distance between them. We use the **hue** to distinguish between items at the respective zoom levels, as described in Section 4. We describe how we dynamically adapt the hues as the user is panning the scene to ensure color discriminability even in a scene with dozens of different entities, and how hues are sub-divided hierarchically as the user zooms in (Section 4). **Luminance** is used to show hierarchical aggregation of quantitative information from lower hierarchy levels (Section 5.1) or to indicate items from lower hierarchy levels to smooth color transitions (Section 5.2). We illustrate our technique with usage scenarios from two domains: molecular visualization and geographic visualization. Finally, we demonstrate the effectiveness of our technique in a user study with professionals and students from the field of molecular biology (Section 7), where they investigated a multi-scale HIV data set.

4. Dynamic Hierarchical Hue Assignment

Initially, our task is to find colors for each visible item so that users can discriminate these items, but also understand their hierarchical grouping. We use hue for this discrimination and visual grouping of items, since it is recommended to use hue for low frequency information [BRT95]. The task of finding suitable hues can be considered as defining “wedges” on an iso-luminant hue circle in HCL space, centered in the grey spot, as visualized in Figure 4, fourth column. Each wedge corresponds to a group of items, and individual items are placed equidistantly within these wedges, as illustrated in Figure 4, top row.

To explain the calculation we define the following terms:

1. α_{max} : The maximum permitted angle between two neighbouring items on the hue circle.
2. β_{max} : The maximum permitted angle between two neighbouring wedges.
3. α : The angle between two neighbouring items on the hue circle. $\alpha = \min(s_\alpha \cdot \alpha_{max}, \alpha_{max})$.
4. β : The angle between two neighbouring wedges. $\beta = \min(s_\beta \cdot \beta_{max}, \beta_{max})$.
5. s_α : The scaling factor for calculating α .
6. s_β : The scaling factor for calculating β .
7. h_{ij} : The hue value of the visible item j in group i .

Figure 5a illustrates α , β , and different groups.

For the initial hue assignment of visible items in n visible groups, we allocate hues with respect to four adjustable parameters: α_{max} , β_{max} , s_α and s_β . α_{max} and β_{max} are a set scalar value. s_α and s_β are used to determine α and β respectively, and they are a parameter in the sense that they are functions that the user can set. An example can be seen in Equation 2 and 3. As the number of visible items may vary, the scaling factors must take the number of items into account and decrease or increase accordingly. The scaling factors must ensure that the items all fit on the hue circle, i.e., no h_{ij} exceeds 360 degrees.

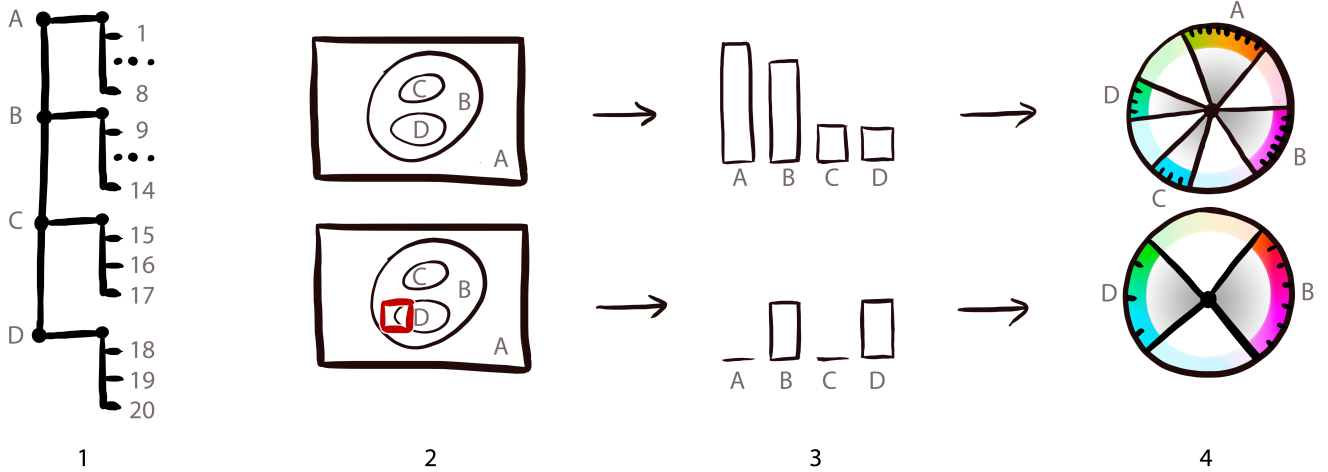


Figure 4: Overview of the view-dependent hue assignment (1) an exemplary two-level hierarchy of four groups (A,B,C,D) and a total number of 20 different items: For the current viewport (2), visibility information is extracted (3) and the color assignment is updated (4).

Through s_α and s_β we can calculate α and β , which gives us h_{ij} :

$$h_{ij} = i \cdot \beta + \left(\sum_{k=1}^{i-1} m_k \right) \cdot \alpha + j \cdot \alpha, \quad (1)$$

where m_k is the number of visible items in group k . Using Equation 1 is the same as placing the items in order along the hue circle. An illustration showing the placement according to Equation 1 can be seen in Figure 5a.

First paragraph after equation 1, first sentence: In the simplest case, the scaling factors are uniformly chosen for adjusting the spacing between items and wedges, respectively. Alternatively, it may be desirable to put more emphasis on the group an item belongs to. Consider Figure 6a, where the spacing between the wedges (β) is considerably large, while the items within a group are assigned quite similar colors, i.e., α is small. In this example, with n being the number of visible groups, s_β and s_α are defined as:

$$s_\beta = \frac{360^\circ}{n}, \quad (2)$$

$$s_\alpha = \frac{360^\circ - n \cdot \beta}{\alpha_{max} \sum_{i=1}^n m_i}. \quad (3)$$

In this case, the items within a group would all be assigned an identical hue value before the hues between groups would become more similar, i.e., the group an item belongs to would still be distinguishable even if items within a group cannot. Of course, the scaling factors can be modified to emphasise groupings as desired. In the treemap example shown in Figure 6c, the number of items and the chosen maximum hue angles are sufficiently small so that α and β are equal to α_{max} and β_{max} respectively. However, if there are a lot of visible items and groups, colors can become very similar and hard to distinguish, such as in Figure 1a. This effect is actually desirable to avoid salt and pepper noise. However, as we zoom in,

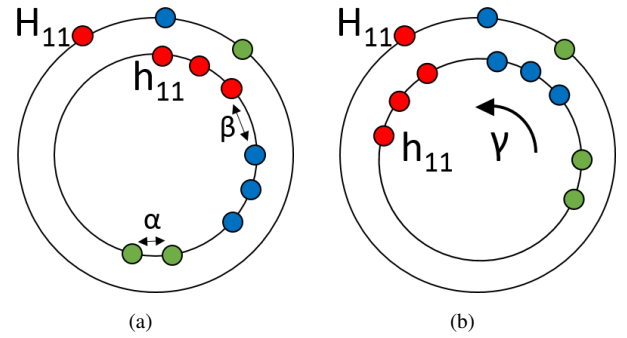


Figure 5: Placement of items on the hue circle. The parents are on the outer circle. The children are on the inner circle. The colors show which items belong together. h_{ij} and H_{ij} are marked for one item. (a) shows the placement of the child items on the inner circle according to Equation 1. α and β are the angles between items and wedges respectively. (b) shows the position of the child items after the rotation by γ according to Equation 5.

we want increasing discriminability (Requirement 1.), while maintaining information about the hierarchy. We therefore introduce a hierarchical view-dependent color adaptation approach that takes the number of visible items into account, and thereby adjusts the scale factors s_α and s_β dynamically, based on the current visibility, while minimizing the difference in hue between child and parent items.

4.1. Hierarchical View-Dependent Color Adjustment

As the user zooms in and explores the visualization, many items may not actually be visible in the current view, as illustrated in the lower row of Figure 4. In this configuration, the available colors becomes underexploited. We therefore recompute and optimize

the color assignment on-the-fly as items appear and disappear. An overview of the mechanism is illustrated in Figure 4.

For every zoom or pan operation by the user, the hues of the ordered set of visible items are reassigned (Equation 1), and the scale factors are re-computed. To explain how, we introduce H_{ij} – the position of the parent of item j from group i . To avoid unnecessarily large color shifts (Requirement 2.) while navigating, we want to minimize the distances between H_{ij} and h_{ij} . In order to achieve this, we introduce a shift γ , i.e., the position of an item on the hue circle becomes $h_{ij} + \gamma$, where h_{ij} is defined through Equation 1. This means that the distance between the position of an item on the hue circle and H_{ij} is equal to $|H_{ij} - h_{ij} - \gamma|$. To find γ , we minimize the sum of squared distances, which we define as G , using a differential approach. If we allow the distances to be weighted, then G is:

$$G = \sum_i^n \sum_j^{m_i} (\omega_{ij}(H_{ij} - h_{ij} - \gamma))^2 \quad (4)$$

where ω_{ij} is a weight factor that can be used to put more emphasis on large items or groups, so that their colors maintain as stable as possible. An illustration of the shift by γ can be seen in Figure 5.

In order to maintain clarity, we want to maintain the equidistant hue distribution of items given by α and β , i.e., the distances between items and wedges respectively. Therefore we apply a simple global rotation (similar to a rigid transformation) instead of shifting individual wedges or items independently. This means G is minimal if γ is equal to the average weighted angular displacement between each visible item's currently assigned hue h_{ij} and its parent H_{ij} . Finding the minimum through a differential approach leads to the following solution:

$$\gamma = \frac{\sum_i \sum_j ((H_{ij} - h_{ij}) \cdot \omega_{ij})}{\sum_i \sum_j \omega_{ij}} \quad (5)$$

The weight can be used to affect how much the color changes for important objects, as their large influence on γ will minimize their shift more than others. For instance, in Figure 6, the geographic area of the items was used as weight factor, which is also encoded by the size of the rectangles in the tree map. As the user zooms from a world overview (Figure 6a) down to the Central American's countries (Figure 6c), the weighted color shift ensures that Mexico, as the largest Central American country, maintains a similar color to its parent (Central America in Figure 6b), while smaller countries, like Panama, have little effect on γ and can have a large difference in color to its parent.

Our continuous solution guarantees that a small change in visible items will lead to a small change in the coloring (Requirement 4.). The position calculation is deterministic, therefore the coloring is consistent and particular views always have the same color assignments (Requirement 5.).

4.2. Hierarchical Colors

While zooming, the visualization may reveal items on lower zoom levels that were initially invisible. Either, the visualization has several discrete hierarchy levels that are switched based on the zoom

factor (such as the treemap example in Figure 6), or the visualization can be continuously zoomed (such as the HIV example in Figure 1). In the latter case, each hierarchy level has to be mapped onto an appropriate zoom factor, so that its associated items are optimally visible. Color blending assures that color transitions are seamless (Section 6).

There are two methods to assign hues hierarchically: The first method is to assign for each visible item on the current hierarchy level the hue independently of its parent color, as denoted in Equation 1. Afterwards, we minimize the items' hue displacements with respect to their parents' hues as described by Equation 5, where H_{ij} is the hue of the parent by the visible item j in the visible group i . This method guarantees that groups of items do not use overlapping hues, yet have a minimized hue shifts from their parents. We used this method for the treemap example (Figure 6) and the zoomable political map (Section 4.4).

The second method assigns colors for each group of items independently. Given a parent hue of the i 'th group H_i , the hue ring is further sub-divided into wedges centered around the parent hue. The effect of this parameter can be seen in Figure 7. The hue angle between the items is α_{max} , by default, and can be scaled down so that all items fit within a pre-defined maximum wedge size. Mind that hue wedges can overlap in this case. This means that items from different groups can be assigned identical hues. While this supports discrimination within groups and leads to minimal hue displacements, it may be difficult to distinguish groups from another. We therefore only use this wedge sub-division approach for groups that are clearly separated through other visual means, such as proteins, which is shown in Figure 8.

4.3. Application: HIV

With improving technology, more and more data across all levels—from atomic to compartment—is being gathered on viruses and cells. To keep up with the progress, new visualization methods were specifically developed to address this challenge, such as by Lampe et al. [LVRH07] or Falk et al. [FKE13] (see Kozlikova et al. [KKL*15] for an overview). Today it is possible to display an ever increasing amount of this data at once, up to and beyond an entire human immunodeficiency virus (HIV) [LMAPV15], with its complete macromolecular composition, which can be explored interactively. Therefore, structural biological models are of particular interest for multi-scale color mapping.

In Figure 1, the different levels of an HIV data set are shown. When zooming into the HIV data set, we start at the compartment level (1a), continue through the protein level (1b), to the domain (1c), secondary structure (1d), and atomic level (1e). A domain of a protein is a part of the protein sequence and is often individually stable and can often be folded. A domain contains secondary structures, such as α -helices and β -sheets. By applying our Cuttlefish method, the individual structures, i.e., the items on this level, are distinguishable at each level. Furthermore, they are grouped by their parent, indicating where one group begins and another ends. For instance, in Figure 1b, individual proteins are assigned slightly different colors, while the hierarchical grouping (capsid hull and interior) is visible. For the lowest hierarchy level,

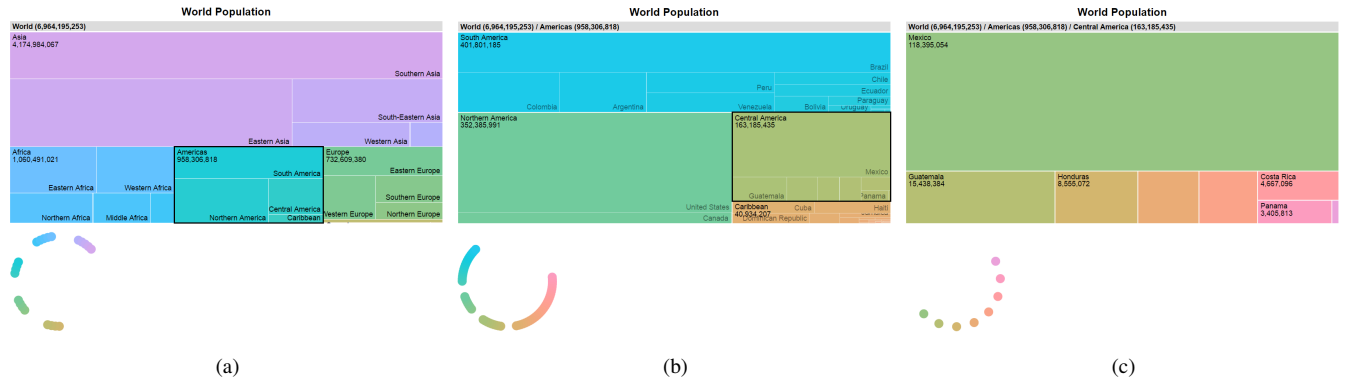


Figure 6: Applying Cuttlefish on a zoomable treemap [Gan16]: (a) The initial world view to (b) Americas and (c) Central America. In this example, we use an α_{max} of 5° , a β_{max} of 30° between wedges, and the countries' areas (encoded by size) as weights to balance the hue shift across scales. The color wheels underneath the tree maps show the assigned colors.

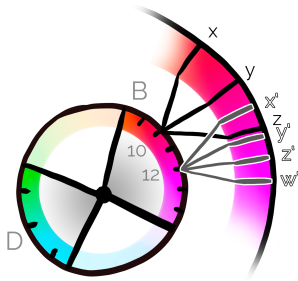
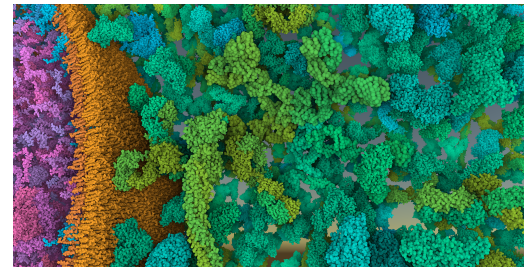


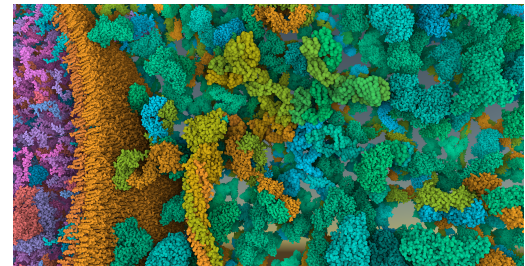
Figure 7: Hierarchical hue assignment by setting child wedges centered at the parent hue.

we use the pre-defined CPK color scheme for atoms to support domain knowledge (see Figure 1e).

To determine the hue assignment, we extract visibility information by analyzing the previously rendered frame. For each protein group, we determine the number of visible protein types m_i . A protein type is visible, if at least one screen pixel is occupied by a protein of type i . We leverage GPU computing in a post-processing operation, in order to compute the visibility of each protein type efficiently. Upon rendering, we generate an additional offline texture which contains, for each pixel, the unique identifier of the rendered protein. There are two GPU buffers which will, for each protein type, store the occupied pixel count and the total number of visible proteins, respectively. Subsequently, in a dedicated compute shader, we iterate through all the pixels, and we increment the pixel counter corresponding to the protein type stored in the video memory. At the same time, we also flag in third dedicated GPU buffer the proteins whose unique IDs are present in the generated texture. This information will allow us to determine the number of visible instances for each protein type. In a second pass and in a dedicated compute shader we then iterate through all the protein instances. For each protein which was flagged as visible in the previous pass, we increase the number of visible instances corresponding to the protein type. Since the computation is done in parallel it is impor-



(a)



(b)

Figure 8: Effect of maximum children wedge size: (a) 60° and (b) 180° . Mind how the domains of different proteins are assigned identical colors in (b).

tant to use atomic intrinsic functions to increment the counters, in order to avoid concurrent access to the same value from two different threads.

4.4. Application: Zoomable Political Maps

Cuttlefish can also be used for geographic visualization. Political maps use categorical colors to distinguish individual countries, states, or districts. In Figure 9, we show a zoomable map with dynamic multi-scale colors. Hues are initially assigned from an overview perspective over Europe. As the user zooms in and countries are clipped from the view, the country colors are re-adjusted

for better discrimination. At a pre-defined zoom level, the map switches from a country to a state representation. Our Cuttlefish algorithm takes care that each state is assigned a unique color, and that the hue shifts from the parent (i.e., country) are minimized.

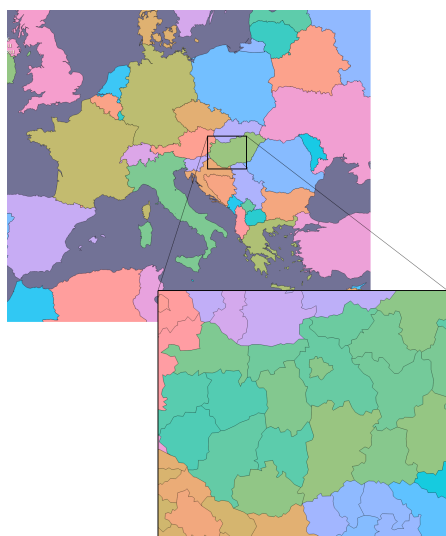


Figure 9: Zoomable political map: closing in from Europe to Hungary.

5. Luminance Modulation

Our dynamic categorical color assignment only affects the hue. This leaves the other two HCL channels available to encode additional information. We show two methods how to encode additional information using these channels: to additionally encode hierarchically aggregated quantitative information (Section 5.1) and to subtly indicate items from a lower hierarchy level (Section 5.2).

5.1. Quantitative Color Coding

When viewing the data at a high level, the user may be interested only in a subset with certain characteristics present on a lower level. Therefore, it may be necessary to propagate the information up to the current level through hierarchical aggregation of quantitative information defined on a lower level.

Quantitative information is defined on the lowest hierarchy level of the data, for instance by items associated with geographic locations. For each zoom level, the number of items is aggregated per displayable unit, such as the number of items per country. For each hierarchy level, the range of aggregated items is then mapped to a luminance range, with a higher luminance representing a higher number of items per unit. In our examples, we use a luminance range of [33,73] (maximum luminance range in HCL is [0,100]).

However, the color saturation (or chroma) for different luminance values depends on the luminance in the monitor's RGB space. For the highest used luminance in our examples, we can achieve the highest possible chroma of around 40, assuming the monitor is calibrated to sRGB. With decreasing luminance, the

chroma – and thereby the radius of our hue circle, as shown in Figure 4, fourth column – decreases. We therefore also linearly interpolate between a chroma range of [22,41] in our example, based on the item's associated quantitative value. This means the Colors are chosen from a cone in HCL space.

5.1.1. Application: Lysine Concentration in HIV

The Lysine amino acid is often used as the target for chemical crosslinking agents (contain reactive ends to join two or more molecules), which can be used to study how different subunits are arranged relative to one another in higher-order assemblies. Lysine is often the target for these crosslinkers, because there often many Lysine amino acids on the surface of typical proteins. Indicating to the user the prevalence of these amino acids, as well as their position is therefore desirable.

The results of the luminance modulation for the HIV data set for different levels are visualized in Figure 10. Figure 10a shows that Lysine is fairly evenly distributed among the virus proteins. When zooming in (Figure 10b), it becomes obvious that lysine is present in the capsid protein, but only within certain secondary structures (as visualized in Figure 10c).

5.1.2. Application: Choropleth Map

For the second example, we extend the zoomable political map from Section 4.4 with quantitative information and thereby create a choropleth map. Classic choropleth maps encode quantitative information for discrete areas using a quantitative color scale, where the hue is usually static, and the luminance is varied linearly or logarithmically with the associated quantitative information.

In our example, we use the political map, as shown in Figure 9, as the foundation, so that the hue is used to differentiate between the individual countries or states. We use luminance and chroma within the ranges described in Section 5.1 above to encode publication counts from a publication database of one of our collaborators. The publication counts are aggregated for each discrete zoom level from the publication records, where each publication record is associated with at least one affiliation and its geographic location, respectively. We linearly map the minimum and maximum publication number of the visible items to the luminance and chroma ranges so that publication activities between political regions can be compared on multiple scales. In Figure 11, for instance, it is noticeable that most publication output in this database is provided by central European countries, and in particular by the capital regions Vienna, Prague, and Bratislava.

5.2. Luminance Modulation of Detail Structures

Luminance can also be used to indicate lower level structures. For example, molecular illustrators sometimes use the luminance channel to indicate structural properties in lower hierarchy levels, as shown in the close up in Figure 2 or in the "Molecule of the Month" [Goo13]. Here, luminance is used to indicate structural information about lower hierarchy levels. This is a reasonable choice, given that hue is known to be an effective channel to encode low-frequency information, while luminance is more effective to encode

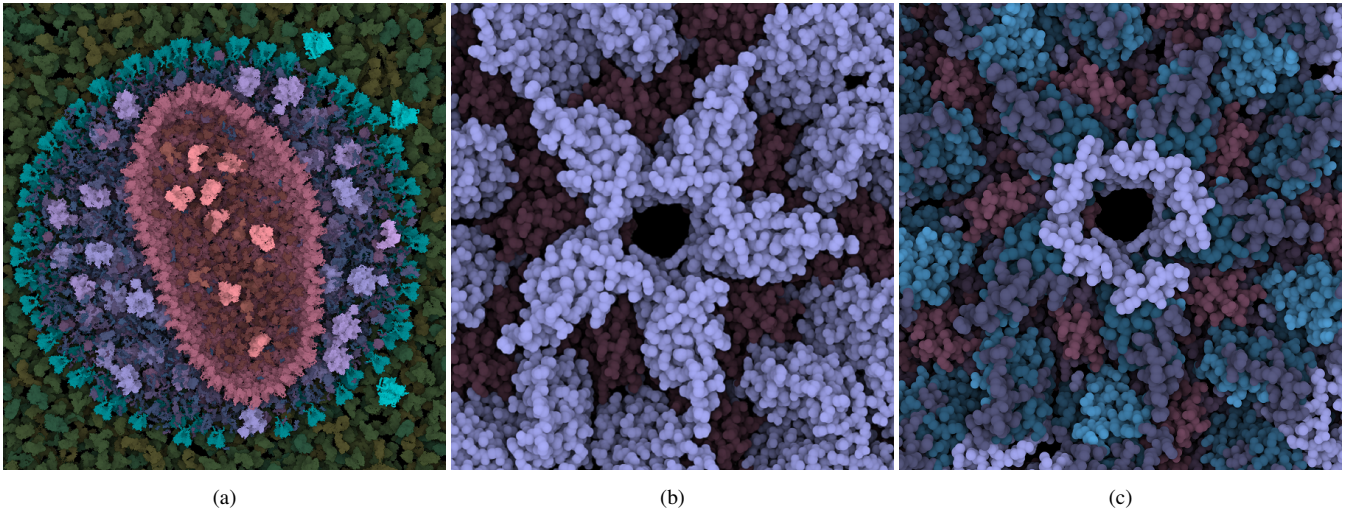


Figure 10: Use of luminance to indicate the presence of the Lysine amino acid. A higher luminance indicates a higher number of amino acids. In (a) the presence in the proteins is shown. In (b) and (c) the concentration of Lysine in a pentamer of the Capsid is shown, domain and secondary structure level respectively.

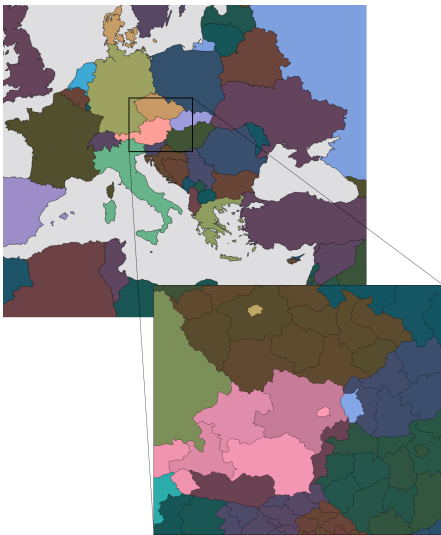


Figure 11: Zoomable choropleth map, where the hue encodes the political entity and luminance and chroma encode the amount of publications produced by the depicted region: zooming from Europe to Austria and its Eastern neighbors.

high-frequency information [BRT95]. In addition, encoding information from an additional zoom level can support orientation when seamlessly transitioning between zoom levels.

For each hierarchy level, we can therefore not only assign unique hues, but also indicate lower level structures with luminance values that are unique within the item they belong to (but not between items) in our HIV application example. We set the base luminance to 73 and modulate this luminance value up to 15, as illustrated in Figure 12. The modulation of protein secondary structures and

atoms is shown in the top and bottom rows respectively. For secondary structures, β -sheets have lowered luminance and α -helices increased luminance. Parts of the protein that have neither are unchanged. In the case of atoms, the luminance is decreased if it is not a carbon atom (Figure 12d), creating an effect similar to the illustrator technique shown in Figure 2. With this small modulation, the luminance channel serves only as a subtle indication of lower-level structural properties, to avoid extensive visual clutter and not to interfere with shading cues.

6. Interpolating Between Levels

To smoothen the transition between zoom levels in continuously zoomable visualization (Requirement 3.), we linearly interpolate the colors in between the levels. The current hierarchy level is calculated by mapping the camera distance to discrete zoom levels. Currently, users have to manually define the mapping distance for each zoom level. In the future, we plan to investigate automatic camera-distance settings, based on the screen size of the items associated with the zoom levels. From the camera distance we are able to compute the current degree of transition between two levels, as well as the hue values associated with these two levels. Finally, the hue of an item is computed by performing a linear interpolation of the hue value of the two levels, as illustrated in Figure 13. In our use case of the HIV data set, the calculation is performed in the fragment shader. The different color components can be treated for each screen pixel individually. This way, information of multiple scales can be seen in a single visualization, as shown in Figure 14.

In the luminance-modulated HIV example (Section 5.2), like the hue, the luminance value is also defined by the distance of the camera to the respective protein structures. However, given an interpolation factor derived from the zooming distance to the pre-defined zoom levels, luminance values are interpolated between the next two hierarchy levels with the current interpolation factor, as illus-

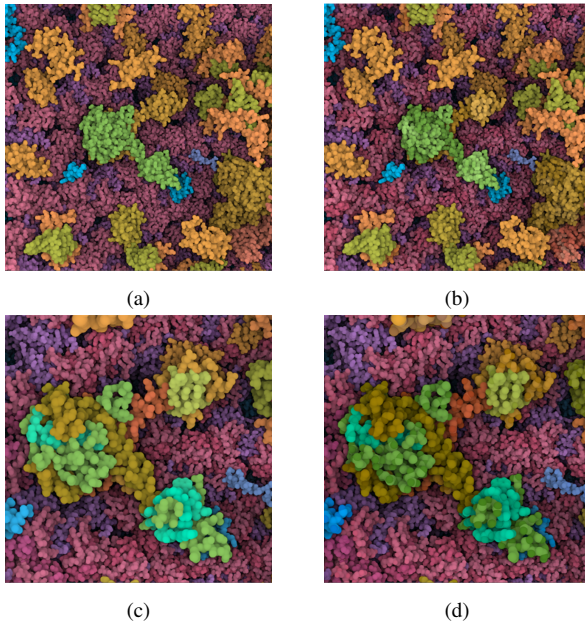


Figure 12: Comparison of domain level (top) and secondary structures level (bottom) without (left column) and with (right column) luminance modulation.

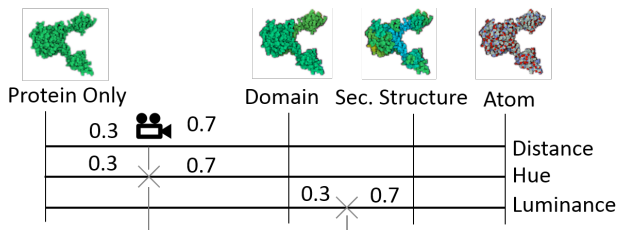


Figure 13: Interpolation of hue and luminance values for structural information: world space distance to the camera defines interpolation factor for hue and luminance offset (Section 5.2).

trated in Figure 13. This way, the hue and luminance encode different hierarchy levels and thereby generate an effect similar as shown in the close up in Figure 2.

7. User Study

To showcase the capabilities of the Cuttlefish color mapping system, we conducted a user study on the HIV data set (see Figure 15). Since interactively explorable multi-scale visualizations of biology models have only become available very recently, there is no comparable approach how to represent biological structures across multiple scales. Dynamic visual discrimination, apart from geometric levels of detail [PJR*14], have not been studied so far in the biological field. We therefore decided for a qualitative evaluation since there is no clear baseline for a comparative lab experiment, in order to answer two research questions:

1. Does the dynamic color mapping support discrimination of protein structures on multiple scales?

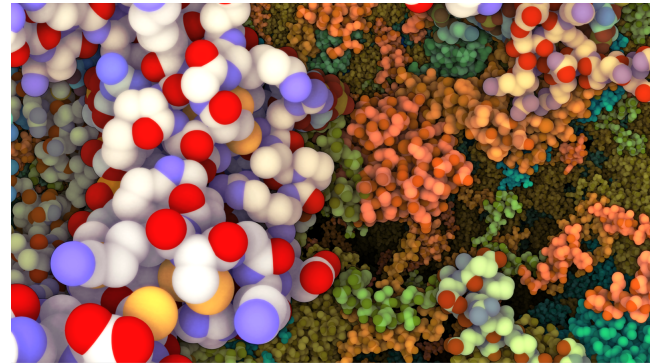


Figure 14: Multiple zoom levels visible simultaneously.

2. Are the dynamic color changes distracting / unpleasant in the exploration process?

To answer these two research questions, we asked five students and professionals in the field of biology to perform two tasks. In the first task (*structure identification task*), users were asked to identify structures on multiple levels of detail. We based our task description on the a previous publicly available description of the HIV capsid at PDB-101 [Goo13]. Based on this description, users were asked to identify the following structures:

1. one of the 12 pentamer capsid proteins,
2. the N-terminal and C-terminal domain of the capsid protein,
3. the alpha helix in the N-terminal domain stabilizing the hexamers / pentamers,
4. the binding site of Cyclophilin A, which is a loop on the surface of the capsid protein with several proline amino acids, and
5. one methionine amino acid within the alpha helix.

The last item was not included in the expert's task description, but was added as a representative task on the amino acid and atom level, respectively. Since we did not color-code the amino acid level explicitly, users were given the hint to identify methionine based on the coloring of its sulfur atom. To assess the performance in the structure identification task, we recorded whether users were able to correctly identify the above listed structures. In the subsequent *free exploration task*, users could freely navigate through the visualization, while thinking aloud. All reported insights were noted. Both tasks were video-taped and followed by a questionnaire and a semi-structured interview. Before the study, users could play around with the tool to get familiar with the navigation, and were also instructed how to toggle the visibility of the protein groups.

With the structure identification task, we could assess whether experts with sufficient knowledge to understand molecular structures without additional text labels are able to identify the above structures using our system. While Cuttlefish provided the necessary discrimination of individual structures in the respective zoom levels, the identification of the structures was only possible through their structural properties (with exception of the methionine amino acid, which was indicated by the yellow sulfur atom). Through the free exploration task, our goal was to assess whether users would notice and be distracted by our dynamic color mapping.

Mind that structural information alone is not sufficient to dis-

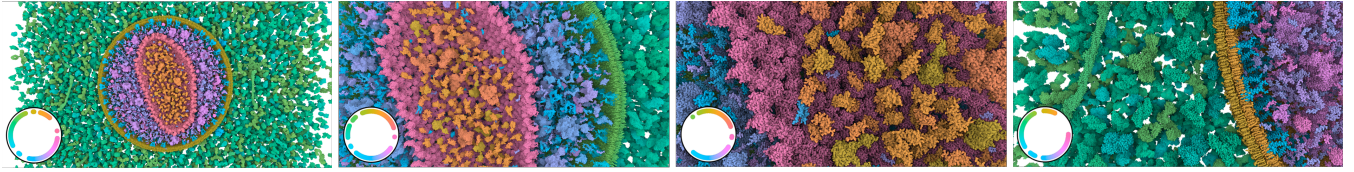


Figure 15: Various snapshots from the multi-scale HIV visualization: As the camera position changes, the discriminability between protein types is optimized. The color palette widget at the bottom left allows us to visualize the distribution of the groups along the hue ring. The hue ring is not intended to be directly interpreted. It is a technical complement to support intuitive understanding.

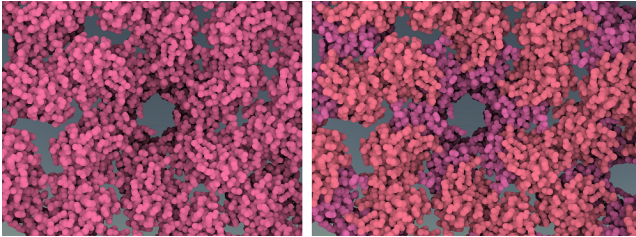


Figure 16: Comparison between static protein coloring (left) and hierarchical color coding using the Cuttlefish color mapping (right) for viewing a capsid protein hexamer on the protein domain level.

criminate structures below the protein level, as shown in Figure 16. In addition, performance measures of such complex tasks are rather hard to compare with only a small number of expert users performing the study.

Therefore we report observations, questionnaire results, and feedback from five experts or students in the field of molecular biology (2 PhD students, one Post-Doc, one pharmacist, and one master student; one female, aged 24 to 31, all with normal or corrected-to-normal vision). While dynamic color mapping is also aimed at the general audience, no such users were involved in the study. The task description for such users would have to be rather long and detailed, likely leading to issues with text comprehension or revealing too much information.

7.1. Results

Table 1 summarizes the performance of users in the structure identification task. Except for user 3 (the pharmacist), all users were able to correctly identify all structures down to the secondary structure level. User 3 mixed up the N and C terminals of the capsid protein, and the rest of the tasks are to some degree based on finding them. With regards to the amino acid, there was more than one kind of amino acid with a sulfur atom, and two users mistook this amino acid for the correct one.

In the questionnaire, users assessed the identification of compartments, but also of proteins, as very easy (see Figure 17). However, the domain and secondary structure identification was rated as much more difficult. This is also reflected in the users' self-reported certainty of the identified structures (Table 1). While the users were quite certain about the identity of a pentamer capsid, once they spotted it, they were least certain about the *alpha*-helices stabilizing the hexamers and pentamers. All users, except for user 3, were

Table 1: Structure identification task: **c** identified correct structure, **p** partially correctly identified structure, **i** identified incorrect structure, **n** nothing found. User-reported certainty shown on a scale of 1 (lowest) to 5 (highest).

Task \ User	1	2	3	4	5
Pentamer	c (5)	c (5)	i (4)	c (5)	c (5)
C-terminal	c (5)	c (4)	i (1)	c (4)	c (4)
N-terminal	c (5)	c (4)	i (1)	c (2)	c (4)
α -helix	c (1)	c (3)	n	c (4)	c (2)
Binding site	c (1)	c (5)	c (5)	c (5)	c (4)
Methionine	c (1)	p (5)	n	p (1)	n

able to identify the C- and N-terminal domains of the capsid proteins. All those users verbally referred to the domains by color. Two users also referred to the α -helices by color.

In the free exploration task, all except for User 3 explored the virus. On average, they spent ten minutes for exploration. Some users reported that they learned something new when exploring the visualization, such as that “*HIV uses [the] host membrane*” and “*how crowded everything is*”. The four users focused on different parts of the virus during their free exploration, such as the membrane (User 1 and 5), the proteins in the matrix (User 1 and 4), and the amino acids of the Cyclophilin A binding site (User 2).

In the questionnaire, most users indicated that they noted the color change (see Figure 17), but the confusion caused by these color changes was rated fairly low. When asked about the changing colors in the post-experiment interview, all users reported that they noticed changing colors when zooming to the atomic level. However, only one user noticed it on all levels, while a second one thought “*something odd*” was occurring on the secondary structure level.

In general, our users all issued the highest possible grade for visual appeal in our questionnaire. In the post-experiment interview, they explained that they considered the tool useful for presenting their research and educating students – which is in line with our research goals. However, they also had suggestions for improvement, such as adding text labels, visually marking the termini of the protein domains, and providing a cartoon representation for secondary structures.

7.2. Summary and Discussion of Results

The performance and feedback of the molecular biologists in our study indicate that the dynamic color mapping supports users in identifying molecular structures on multiple scales. Users were

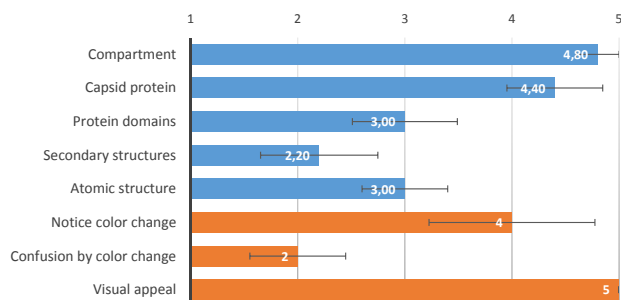


Figure 17: Average and standard error of responses on a five-point Likert scale for questions concerning the ease of identification of structures (blue), as well as color changes and visual appeal (orange).

equally successful identifying one of the capsid proteins forming a pentamer, and identifying the two domains of the capsid protein. The pentamer capsid protein differed only in shape from the more frequent hexamer capsid proteins, while the two different domains of the capsid proteins were encoded by color. The users reported lower certainty and higher task difficulty for the identification of the domains. This is an indication that identification by structure is easier than identification by color alone – yet, color can be used if no strong structural cues are available, as in our protein domain example.

On the secondary structures level, the identification rate was similarly high, but users were quite uncertain about their findings and reported a high task difficulty. On this level, α -helices and β -sheets were discriminated by color, similar to Figure 12. However, the particular α -helix mentioned in the task description could only be identified as a helix by shape. The request for an alternative cartoon representation for identifying structures on this level shows the limits of multi-scale color mapping without adapting the structural representation. In the future, it will be important to explore combinations of semantic zooming comprising both, structural and color information for multi-scale biological visualizations.

User feedback shows that our system's dynamic color changes do not interfere with the users' workflow. While protein domains and secondary structures were mainly referred to by their color, only few users noticed the color changes (except at the atomic level) and none found them confusing. All users found the visualization highly appealing and useful for presentation and education purposes. We therefore conclude that the Cuttlefish color mapping for dynamic multi-scale visualizations is a valuable extension to multi-scale molecular visualization. It does not cause any notable distraction and allows for an efficient and pleasing visual encoding of protein substructures.

8. Conclusion

The work presented here is a method that is capable of visualizing a hierarchical data structure on different levels in a coherent and cohesive manner using color. The hierarchical structure of the coloring technique, along with the visibility based variation in color allows users to navigate and inspect parts of the visualization without

getting disorientated while being subtle. Altering the color based on semantic zooming changes the meaning of color coding. Instead of discriminability for any situation, semantic color changes allow items to be clustered or separated as necessary. This is achieved while maintaining the connection between items on different levels in a logical manner via the hierarchical nature of the coloring. Furthermore the coloring scheme is capable of showing items on each level in a visually distinct manner. The user study was performed with a research prototype and shows that experts can find information in an HIV dataset on each level, while not being distracted by the dynamic color mapping. In the future, it will be interesting to formally investigate the usefulness of our technique in non-biological areas as well.

Acknowledgments

This project has been funded by the Vienna Science and Technology Fund (WWTF) through project VRG11-010 and supported by EC Marie Curie Career Integration Grant through project PCIG13-GA-2013-618680 and the Austrian Science Fund FWF through project T 752-N30.

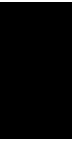
References

- [BL07] BLANCH R., LECOLINET E.: Browsing Zoomable Treemaps: Structure-Aware Multi-Scale Navigation Techniques. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1248–1253. doi:10.1109/TVCG.2007.70540. 2
- [BRT95] BERGMAN L. D., ROGOWITZ B. E., TREINISH L. A.: A rule-based tool for assisting colormap selection. In *Proceedings of the 6th conference on Visualization '95* (1995), IEEE Computer Society, p. 118. 2, 3, 8
- [BSM*15] BERNARD J., STEIGER M., MITTELSTÄDT S., THUM S., KEIM D., KOHLHAMMER J.: A survey and task-based quality assessment of static 2d colormaps. In *IS&T/SPIE Electronic Imaging* (2015), International Society for Optics and Photonics, pp. 93970M–93970M. 2
- [CWM09] CHUANG J., WEISKOPF D., MÖLLER T.: Hue-preserving color blending. *Visualization and Computer Graphics, IEEE Transactions on* 15, 6 (2009), 1275–1282. 2
- [EF10] ELMQVIST N., FEKETE J.: Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics* 16, 3 (2010), 439–454. doi:10.1109/TVCG.2009.84. 2
- [FKE13] FALK M., KRONE M., ERTL T.: Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 195–206. 5
- [FWR99] FUA Y.-H., WARD M. O., RUNDENSTEINER E. A.: Hierarchical Parallel Coordinates for Exploration of Large Datasets. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years* (Los Alamitos, CA, USA, 1999), VIS '99, IEEE Computer Society Press, pp. 43–50. URL: <http://dl.acm.org/citation.cfm?id=319351.319355>. 2, 3
- [Gan16] GANESHV: Zoomable Treemap Template. <http://bl.ocks.org/ganeshv/6a8e9ada3ab7f2d88022>, 2016. [Online; accessed 13-December-2016]. 6
- [Goo13] GOODSELL D.: Pdb-101 molecule of the month: Hiv capsid. <http://pdb101.rcsb.org/motm/163> (2013). [Online]. 7, 9
- [HB03] HARROWER M., BREWER C. A.: Colorbrewer.org: an online tool for selecting colour schemes for maps. *The Cartographic Journal* 40, 1 (2003), 27–37. 2

- [Hea96] HEALEY C. G.: Choosing effective colours for data visualization. In *Visualization '96. Proceedings.* (1996), IEEE, pp. 263–270. [2](#)
- [Iha03] IHAKA R.: Colour for presentation graphics. In *Proceedings of DSC* (2003), p. 2. [2](#)
- [KKL*15] KOZLIKOVA B., KRONE M., LINDOW N., FALK M., BAADEEN M., BAUM D., VIOLA I., PARULEK J., HEGE H.-C., ET AL.: Visualization of biomolecular structures: state of the art. In *Eurographics Conference on Visualization (EuroVis)-STARs* (2015), The Eurographics Association. [5](#)
- [LMAPV15] LE MUZIC M., AUTIN L., PARULEK J., VIOLA I.: cel-view: A tool for illustrative and multi-scale rendering of large biomolecular datasets. In *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine* (2015), VCBM '15, Eurographics Association, pp. 61–70. [1](#), [5](#)
- [LSS13] LEE S., SIPS M., SEIDEL H.-P.: Perceptually driven visibility optimization for categorical data visualization. *Visualization and Computer Graphics, IEEE Transactions on* 19, 10 (2013), 1746–1757. [2](#)
- [LVRH07] LAMPE O. D., VIOLA I., REUTER N., HAUSER H.: Two-level approach to efficient visualization of protein dynamics. *Visualization and Computer Graphics, IEEE Transactions on* 13, 6 (2007), 1616–1623. [5](#)
- [MBS*14] MITTELSTÄDT S., BERNARD J., SCHRECK T., STEIGER M., KOHLHAMMER J., KEIM D. A.: Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis. In *EuroVis - Short Papers* (2014), The Eurographics Association. [3](#)
- [Med16] MEDIALAB: i want hue. <http://tools.medialab.sciences-po.fr/iwanthue/> (2016). [Online]. [2](#)
- [MSK14] MITTELSTÄDT S., STOFFEL A., KEIM D. A.: Methods for compensating contrast effects in information visualization. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 231–240. [3](#)
- [PJR*14] PARULEK J., JÖNSSON D., ROPINSKI T., BRUCKNER S., YNNERMAN A., VIOLA I.: Continuous levels-of-detail and visual abstraction for seamless molecular visualization. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 276–287. [9](#)
- [RCS11] RCSB PDB: The structural biology of hiv. <http://pdb101.rcsb.org/learn/resource/the-structural-biology-of-hiv-flash> (2011). [Online]. [2](#)
- [SBT*15] STEIGER M., BERNARD J., THUM S., MITTELSTÄDT S., HUTTER M., KEIM D. A., KOHLHAMMER J.: Explorative analysis of 2d color maps. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2015)* (2015), pp. 151–160. [2](#)
- [STH03] STOLTE C., TANG D., HANRAHAN P.: Multiscale visualization using data cubes. *IEEE Transactions on Visualization and Computer Graphics* 9, 2 (Apr. 2003), 176–187. [doi:10.1109/TVCG.2003.1196005](https://doi.org/10.1109/TVCG.2003.1196005). [1](#)
- [TdJ14] TENNEKES M., DE JONGE E.: Tree colors: Color schemes for tree-structured data. *Visualization and Computer Graphics, IEEE Transactions on* 20, 12 (2014), 2072–2081. [2](#), [3](#)
- [WGM*08] WANG L., GIESEN J., McDONNELL K. T., ZOLLIKER P., MUELLER K.: Color design for illustrative visualization. *Visualization and Computer Graphics, IEEE Transactions on* 14, 6 (2008), 1739–1754. [2](#)
- [WLW*16] WALDIN N., LE MUZIC M., WALDNER M., GRÖLLER E., GOODSSELL D., LUDOVIC A., VIOLA I.: Chameleon - Dynamic Color Mapping for Multi-Scale Structural Biology Models. In *Eurographics Workshop on Visual Computing for Biology and Medicine* (2016), Bruckner S., Preim B., Vilanova A., Hauser H., Hennemuth A., Lundervold A., (Eds.), The Eurographics Association. [doi:10.2312/vcbm.20161266](https://doi.org/10.2312/vcbm.20161266). [3](#)

PAPER

D



Flicker Observer Effect: Guiding Attention Through High Frequency Flicker in Images

Flicker Observer Effect: Guiding Attention Through High Frequency Flicker in Images

N. Waldin^{†1}, M. Waldner¹, I. Viola¹¹TU Wien

Abstract

Drawing the user's gaze to an important item in an image or a graphical user interface is a common challenge. Usually, some form of highlighting is used, such as a clearly distinct color or a border around the item. Flicker can also be very salient, but is often perceived as annoying. In this paper, we explore high frequency flicker (60 to 72 Hz) to guide the user's attention in an image. At such high frequencies, the critical flicker frequency (CFF) threshold is reached, which makes the flicker appear to fuse into a stable signal. However, the CFF is not uniform across the visual field, but is higher in the peripheral vision at normal lighting conditions. Through experiments, we show that high frequency flicker can be easily detected by observers in the peripheral vision, but the signal is hardly visible in the foveal vision when users directly look at the flickering patch. We demonstrate that this property can be used to draw the user's attention to important image regions using a standard high refresh-rate computer monitor with minimal visible modifications to the image. In an uncalibrated visual search task, users could in a crowded image easily spot the specified search targets flickering with very high frequency. They also reported that high frequency flicker was distracting when they had to attend to another region, while it was hardly noticeable when looking at the flickering region itself.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Computer Graphics]: User Interfaces—Evaluation/methodology

1. Introduction

Many types of media and applications use mechanisms to attract the user's attention. Examples are advertisements, desktop notifications, visualizing responses to human interaction in interactive systems, and guiding the user's attention along a narrative through a dynamic scene. In highly complex scenes with a lot of colors, heterogeneous shapes, motion, and audio, the entire content often needs to be visually compressed – for instance, darkened [KMF05], blurred [KM02], or scaled down [Fur86] – to make an item of interest stand out effectively. This may be undesirable due to information loss [ZWSK97], distorted perception of the scene [GF04], or simply because users feel disrupted when the entire content is modified.

In order to gently attract the user's attention, very subtle techniques are employed, for instance modulation of visual saliency [MB16, KV06, VMFS11]. However, subtle techniques, in turn, can be easily missed. What would be desirable is a technique that effectively catches the user's attention only when the attention is cur-

rently directed somewhere else, and ceases to be noticeable when directly looking at it.

In the past, such an effect was achieved by using an eye tracker to detect whenever a user initiated a saccade towards the image's area of interest and to remove the modulation [BMSG09]. However, an eye tracker may restrict the user's freedom of movement and limits the application to a single user. In this paper, we describe a method for attracting an observer's gaze using flicker on high frequency monitors – which we call the *flicker observer effect*. These monitors are becoming more widespread nowadays to provide high refresh-rate technology or the possibility to interpolate between movie frames in order to decrease motion blur. We thereby make use of the fact that sensitivity to flicker varies across the retina [Tyl87], with sensitivity being higher in the periphery. This means that at a certain frequency, flicker can only be perceived in the visual periphery, but fuses to a stable signal when the blinking image area is directly looked at. However, whether or not a user can detect flicker depends on several other factors than the flicker frequency, such as size and luminance [Dav12]. We therefore performed multiple experiments to investigate the suitability of high frequency flicker to efficiently and non-distractingly attract

[†] nwaldin@cg.tuwien.ac.at

the user's attention in images. As a result, our contributions are as follows:

- Determined through a psychophysics experiment, we present target region size and luminance ranges for targets flickering with 60–72 Hz so that the desired flicker observer effect can be achieved.
- We demonstrate that, using these settings, flickering items indeed can be reliably detected in the visual periphery, but are rarely seen in the foveal vision.
- We showcase the usefulness of the flicker observer effect on a visual search task in a complex scene. We show how to modify the images to integrate the flicker observer effect and report subjective feedback by 14 observers, which indicates that the flicker observer effect is highly effective, yet hardly noticeable – and therefore minimally disturbing – when directly looking at the flickering target.

2. Related work

Using some sort of highlighting or “attention retargeting” technique [MB16] to direct the user's attention to semantically important image regions is a common task in various domains, such as user interface design [ZWSK97, BWC03, GBM07, HBW08], visualization [War12, Rob11], or augmented reality [VMFS11]. Highlighting of important scene elements can deepen the observers' insights by strengthening their engagement [HE12] and improving recall [Low03]. We will first give a concise overview of existing highlighting techniques and then present highlighting techniques using flicker.

2.1. Highlighting Techniques

A large variety of highlighting techniques have been used in graphical interfaces, visualizations, or images to effectively guide the user's attention. Taken literally, a highlight effect can be achieved by making the target brighter (or, in turn, by making the surrounding context darker [KMF05]). Others use distinct colors [SOK*16], spatial distortions [Fur86], artificially added leader lines [HBW08], or blur less important elements [KMH02] to make elements of interest visually stand out. A less common form of highlighting involves the usage of motion [BWC03] or stereoscopic effects [AHKMF11].

While a common goal of highlighting is to make elements of interest clearly distinct from their context [War12], others try to achieve subliminal attention guidance by using very subtle highlighting methods. For instance, subtle modulations of image features to selectively increase or decrease the local saliency were used to direct the user's attention in videos [VMFS11] and volume visualizations [KV06]. In contrast, the goal of our flicker observer effect is not to be subliminal. Our intention is rather to keep the visible modifications of the image to an absolute minimum, while still generating a strong attention guidance effect.

2.2. Highlighting Using Flicker

While there are only a few attempts at using flicker to attract the observer's attention, they can be sorted into three groups. The first

group highlights items of interest by modulating their luminance with a constant amplitude and a constantly low frequency around 1 to 3 Hz [BWC03, WB04, HBW08, GBM07]. Depending on the chosen amplitude and frequency, users described the use of flicker as attractor either as disturbing or as rather ineffective.

To reduce the nuisance introduced by stable flicker, the second group uses what can be described as a decaying flicker. The flicker is initiated with a high frequency and amplitude and quickly decays into a smooth pulsation using a lower frequency and amplitude [WLMB*14]. This way, the attention is effectively attracted at signal onset, but the signal is quickly turned into a clearly visible yet not alerting visual indication. However, balancing the highlight strength and subtleness is also always a compromise – either in terms of signal effectiveness or in terms of visual quality.

Another method to reduce the discomfort when using flicker is to make it gaze-dependent. *Subtle gaze direction* (SGD) uses an eye-tracker to turn off the flicker when the user initiates a saccade towards the flickering region of interest [BMSG09]. Studies have demonstrated the effectiveness of SGD for static image regions [BMSG09] as well as narratives through artwork [MBS*12] without users even noticing the flickering. Further research indicates that this kind of gaze manipulation also increases memory recall [BMC*12]. SGD is barely noticeable, but requires a setup with an eye-tracker – something that is not widely available, limits the user's freedom, and can only be used for a single observer.

In this paper, we use a physiological property of the human visual system to achieve the advantages of SGD without having to use a restricting eye tracker. In an experiment we show that we can achieve a similar effect as SGD – namely, attracting the user's attention in the visual periphery and “hiding” the attractor for the foveal vision – without even knowing where the user is currently looking.

3. Physiological Background

Davson describes flicker as a sensation “*evoked when intermittent light stimuli are present to the eye*” [Dav12]. Flicker is an interesting visual feature since regular flickering is rarely employed in visual scenes, such as movies, websites, or computer applications. It could be shown empirically that blinking targets can be easily discriminated from moving distractors [POT08]. This implies that flicker is attracting the observers' attention even when watching a dynamic scene. In human-computer interaction studies, however, it has been shown that flicker is not only effective, but also considered annoying [GBM07, HBW08].

When increasing the frequency of flicker, it “fuses” for the observer and becomes a continuous signal. The frequency at which the signal is perceived as continuous is called “*critical fusion frequency*” (CFF) [Dav12] or “*flicker fusion threshold*”. The CFF is not stable, but rather depends on numerous factors, such as fatigue [Dav55] and training [SNHW05] of the observer, amplitude and luminance of the flickering target [Dav12], and its size (*Granit-Harper Law*) [RR88]. Upon reaching the CFF, the brightness of the flickering (yet steadily perceived) target is the mean of the brightness of one flicker cycle (*Talbot-Plateau Law*) [Tal34], but not necessarily at even higher frequencies.

The CFF additionally depends on the eccentricity of the visual field at which it is perceived. In a bright environment, the CFF increases with eccentricity [Fuk79, TH90, TH93, Sid97, HV33]. In other words: a high frequency flicker might be perceivable in the peripheral visual field, but the flicker fuses in the foveal area (i.e., when directly looking at it). In the fovea, the CFF can range between <10 Hz and ~45 Hz, depending on the retinal illuminance (*Ferry-Porter Law*). In the periphery, CFF up to 60 Hz or 70 Hz have been observed [TH93]. This effect is well-known from old CRT monitors whose 60 Hz refresh rates could be perceived mainly in the peripheral vision [Sid97, Far86].

Bauer et al. [BCP*09] showed that a 30 Hz flicker signal could be more easily spotted than a 50 Hz flicker signal, but that the orientation effect was stronger for the 50 Hz flicker signal. However, in their experiment, targets were arranged on a small circle, covering 6° of the human visual field. Eccentricity effects on CFF were obtained for much larger visual angles [TH93]. For instance, Sidebottom observed a large effect for a 30° viewing angle [Sid97].

Tyler [Ty185] measured temporal-frequency characteristics as a function of retinal location, and found that the temporal-frequency limits increase linearly by a factor of two from the fovea to 45° of eccentricity. Modern monitors cover a visual angle of 60° and more. While there was an attempt to use high frequency flicker at 50 Hz to try and attract attention [MB14], it was unsuccessful. Users had been instructed to determine whether the flicker was on the left or right side of an image they presented. The flicker was in an area that had lower contrast. The authors did not detect any effect by the flicker. However, in contrast to us, they used a 21 inch CRT monitor with a resolution of 1024 × 768 and a chin rest 70 cm away, while we use a 27 inch LC monitor with a LED backlight and a resolution of 2560 × 1440 and a chin rest 50cm away, allowing for larger visual angles along with a significantly higher resolution. While a display device being able to generate attention-attracting flickering (sub-)images with 75 Hz has been proposed in the past [Eva92], there has been – to the best of our knowledge – no empirical demonstration that the desired flicker observer effect actually works on moderately large monitors, and how users experience this kind of attention guidance.

4. Overview

To explore the applicability of high frequency flicker for guiding the user's attention in images, we split our research into two major blocks. First, we investigate whether we can find parameters to generate our anticipated flicker observer effect – namely that we can attract the users' attention letting an item flicker in the visual periphery, while going unnoticed when triggering the flicker in the foveal vision – in a fully controlled, artificial scene.

We first performed a psychophysics experiment to determine the size ranges and necessary luminance matchings for targets flickering with 60 Hz or 72 Hz (Section 5). Subsequently, we used these personalized settings and performed an experiment to test whether flickering targets in the peripheral field of view can be detected more reliably than targets flickering in the foveal vision (Section 6).

Second, we demonstrate the effectiveness and usefulness of the

flicker observer effect to attract the users' attention in images through a use case. 14 users were asked to spot characters in a crowded comic scene from the famous "Where's Waldo" series and report their subjective impressions of the flicker observer effect (Section 7). We present further potential applications of the flicker observer effect and describe how images can be modified to smoothly integrate this effect (Section 8).

5. Experiment: Flicker Fusion Parameters

We conducted a psychophysics experiment to investigate two parameters influencing the perceived stimulus intensity and visual appearance of an item flickering with a high frequency: size and luminance. The amplitude was always set to maximum (i.e., full luminance range from black to white). We only investigated achromatic flicker since chromatic flicker has been shown to be less effective in the past [BMSG09]. The frequency was set to either 60 or 72 Hz, i.e., the refresh rate of the monitor was 120 or 144 Hz. As default, we used 60 Hz. However, people can detect flicker to varying degrees, and some of our users (3 out of 14) performed exceptionally well. In these cases, the user could detect flicker in the center at below 2° degrees. For these three users, we used 72 Hz throughout the entire experiment. Since the measurements obtained for those participants were not outliers, we included them into our analysis.

The goal of this experiment was to find high frequency flicker settings so that the flicker observer effect could be achieved, and to compare these settings across the participants. In particular, we were interested in finding the following settings for simple circular targets:

- A luminance offset that needs to be applied to the non-flickering circles so that their brightness is perceived as equal to the flickering circle.
- The maximum size of an item in the foveal vision so that the circle's flickering just cannot be perceived in the peripheral regions of the circle.
- The minimum size of a circle in the peripheral vision so that users can just perceive the flickering.

The first step is necessary since it has been shown that the *Talbot-Plateau Law* does not necessarily apply for very high flicker frequencies, and that the sensation magnitude of brightness may be inverse to the flicker period [NB64]. This observation was also confirmed in our early pilot experiments, where the brightness of the high frequency flicker circle was perceived as clearly higher than of the remaining circles by all observers. We therefore included the luminance step to our experiment to compensate for the potential brightness differences between flickering and non-flickering image regions.

To compare flicker sensitivity between the foveal and peripheral vision, we defined three target eccentricities. Thus, the experiment was split into multiple blocks:

1. *Brightness Matchings (BM)* to find the luminance offset so that the brightness of the non-flickering half of a circle is perceived equal to the flickering circle half.
2. *Maximum size (MaxS)* to find the largest size of a circle in the foveal vision so that the flicker just cannot be perceived.

3. *Minimum size* to find the smallest size of a circle in the peripheral vision so that the flicker just can be perceived. This step was divided into two sub-blocks:

- a. *Minimum size in close periphery (MinSCP)* to find the smallest size of a just visibly flickering circle in the peripheral vision, close to the foveal vision. The visual angle between the center of the monitor and the nearest point of the circle is 7.8° . This angle allows the use of a stimulus of up to 1.5 times the fovea [Est10] later.
- b. *Minimum size in far periphery (MinSFP)* to find the smallest size of a just visibly flickering circle in the peripheral vision, with the circle edge at the vertical boundaries of the monitor at the vertical axis (up to 37° visual angle) with equal distance along the horizontal axis.

We only roughly sub-divide the visual field into three target eccentricity regions to find a size threshold so that the flicker is still clearly perceivable in the peripheral vision, but does not exceed the less sensitive foveal vision. Previous research suggests that the flicker sensitivity is higher in the far periphery than in the close periphery [Sid97, Far86]. However, our goal is to attract the attention in the entire peripheral vision to encourage users to directly look at the region of interest. Therefore, we added the presumably less sensitive close periphery block as second peripheral condition.

Users performed the blocks always in the order as listed above. This way, the brightness matching could be applied to all subsequent blocks.

5.1. Apparatus

The monitor used was a 27 inch Asus PG278Q, calibrated to sRGB with the D65 illuminant using the i1 Display Pro. The refresh rate was set either to 120 Hz or 144 Hz, if the user could detect flickering of circles smaller than 2° visual angle. The room did not allow any daylight, and the ceiling lights were on at the highest setting. The users' heads were fixated on a chin rest 50cm from the monitor. The resulting visual angle of the monitor was a total of 62° from left to right and 37° from top to bottom.

5.2. Stimuli

In all blocks, the stimuli were composed of one or four circles on a black background. In the BM block, the user was presented with two half-circles next to each other, i.e., they formed a complete circle. The diameter of the circle was set to 3.2° of visual angle. One half of this circle flickered, the other did not. The flickering half-circle was modulated between the extreme RGB-values [255,255,255] (white) and [0,0,0] (black) to achieve the maximally possible flicker amplitude on the monitor. The non-flickering half-circle was initially white.

In the MaxS-block, we showed a single circle in the center of the screen (see Figure 1a). As we used an interleaved staircase procedure, as described further below, we had to choose two initial sizes for this block. The initial sizes of the circles were set to 5.8° and 1.9° of visual angle. The circle was flickering between white and black to achieve maximum amplitude.

For the MinSCP- and MinSFP-blocks, we used the same initial

circle diameters as in the MaxS-block. In these blocks, four circles were shown – two along the horizontal axis, left and right of the center, and two above and below. In both blocks, all four circles were placed equally far away from the center fixation cross. In the MinSCP-block, the distance of the circle boundary to the screen center was fixed to 7.8° (see Figure 1b). In the MinSFP-block, the top and bottom circles were always touching the screen edge (see Figure 1c). This means that, as the user adjusted the size of the circle, the distance to the screen center was varied. In each trial, one randomly selected circle out of the four was flickering. As in the MaxS-block, the target circle was flickering between white and black. The non-target circles were shown in the gray value found in the BM-block.

For all blocks, users were asked to keep their gaze fixated to the center of the screen and not to move their heads. In the MinSCP- and MinSFP-blocks, a fixation cross was rendered to the center of the screen.

5.3. Task and Procedure

In the BM-block, we used a method-of-adjustment to find the optimal luminance offset. Users could adjust the brightness of the non-flickering circle half by scrolling the mouse wheel to increase or decrease the brightness by as little as one RGB step, i.e., $1/255$. If they desired, they could also use the up and down arrow keys to make very small adjustments. However, only one user used the arrow keys. The luminance value of the non-flickering circle half was stored when the user was finished.

In the MaxS and MinS-blocks, interleaved staircase procedures were used. In each trial, users had to press a mouse button when they perceived flickering in one of the circles, and the “f”-key if not. After pressing a mouse button (i.e., positive response), the size of the circles was decreased, after pressing the “f”-key, it was increased. The step size altered the radius of the circle and was decreased over the course of the experiment, starting with $1/20$ of the height of the screen and decreasing to $1/200$ of the screen after four trials, after which it remained constant. Thereby, we left the distance of the circle boundary to the screen center (in the MinSCP block) and screen edge (in the MinSFP block), respectively, constant. This means that the center of the circle was shifted whenever the diameter changed.

The number of trials was dependent on the user's responses. In total, we continued to show the stimuli until we, for each of the two interleaved staircases, gathered ten responses that differed from the previous one (five “reversals” from “not perceived” expressed by the “f”-key to “perceived” expressed by a mouse click, and five reversals from “perceived” to “not perceived”). The size threshold per block was then determined by averaging the size values for all trials where the user's response changed from “perceived” to “not perceived”.

In summary, in the MaxS and MinS-blocks, the following procedure was used:

1. show central fixation cross for 1 second,
2. show stimulus for 1 second,
3. show blank screen with response request until a mouse button or the “f” key has been pressed,

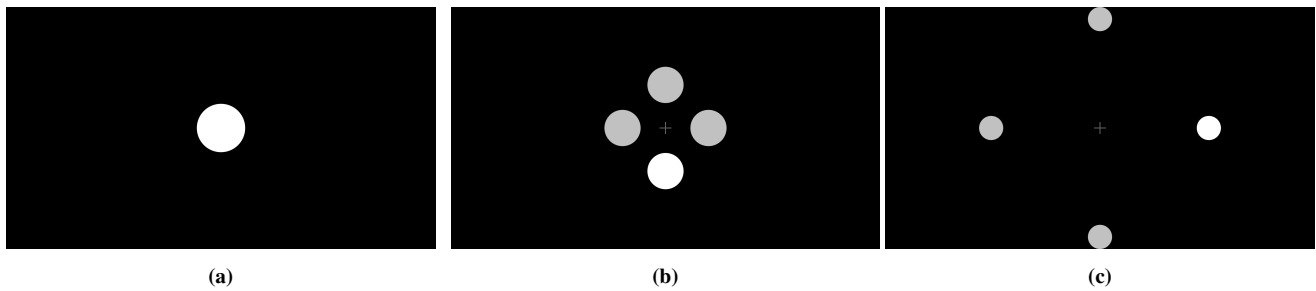


Figure 1: Stimuli of the flicker fusion parameters experiment for the MaxS (a), MinSCP (b), and MinSFP (c) conditions. The flickering circle is indicated by higher luminance. During the experiment, the luminance of the target was adjusted so that the perceived brightness was equivalent for all circles.

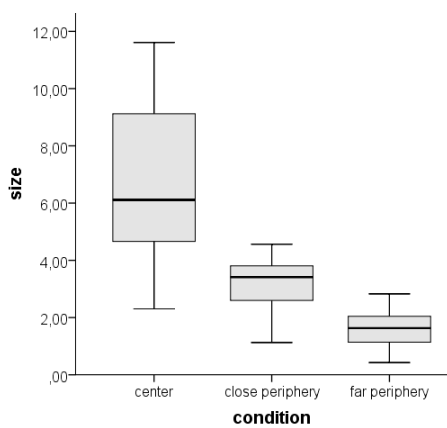


Figure 2: Box plots of maximum visual angles in degrees obtained for the center circle (MaxS) and minimum visual angles obtained for the close (MinSCP) and far periphery (MinSFP) (boxes for first to third quartile and whiskers up to 1.5 IQR).

4. show inverse stimulus image (gray background with black circles) for 100 milliseconds to avoid after-images,
5. adjust circle size according to staircase protocol and repeat, or stop if ten reversals per staircase were recorded after reaching the minimum step size change.

5.4. Subjects

We recruited 14 volunteers from a local university (13 males, 1 female, aged 28 to 54, average 33), each with normal or corrected-to-normal vision and naive to the purpose of the experiment. Six users rarely play computer games, three play weekly, and five daily. All users use a computer on a daily basis.

5.5. Results

For the BM-block, users applied a considerable luminance offset to the non-flickering circle half to achieve equal perceived brightness. On average, the RGB values were [189,189,189] (which corresponds to a luminance value of 76.5 in CIEL*a*b*), with a standard deviation of 4.7. This rather small standard deviation indicates

that the perceived brightness differences were fairly consistent between users.

We performed a repeated-measures ANOVA to compare the size thresholds between the center, the close and the far periphery. The obtained size thresholds are significantly different from each other ($F_{2,26} = 36.319, p < .001$). Bonferroni-adjusted post-hoc comparisons showed that the MaxS-block leads to significantly larger circles (6.55° visual angle, on average) than the two MinS-blocks (3.13° and 1.60° , respectively). As visualized in Figure 2, the resulting MaxS-diameters ranged from 2.31° to 11.61° , while minimum diameters for the peripheral circles were ranging between 1.13° and 4.56° in the close periphery and 0.43° and 2.83° in the far periphery. Mind that these visual angles are given as seen from the user's perspective, focusing to the center of the screen. This means that the effectively seen visual angle of a circle with equal screen size is getting smaller towards the periphery. To find the potential size ranges of flicker circles, we therefore compare the obtained size thresholds in screen pixels. For each user, the usable size range for the flicker observer effect is defined as $[\max(\text{MinSCP}, \text{MinSFP}), \text{MaxS}]$. In Figure 3, we juxtapose these size ranges for all participants, where the usable size range is the difference between the blue and the orange line. As we can see, there is no complete overlap between the resulting size ranges of the users. This means that the effectiveness of the flicker observer effect also depends on the user – and potentially even the personal conditions of the user, as suggested in previous CFF-research [Dav55, SNHW05].

6. Experiment: Flicker Detection in Fovea and Periphery

The purpose of this experiment was to test the underlying hypothesis of the flicker observer effect: *A target flickering with high frequency (60-72 Hz) can be spotted much more reliably in the peripheral vision than in the foveal vision.* Not being able to detect whether or not a target that is currently being fixated is flickering would imply that the distraction introduced by the flicker is negligible.

To verify this hypothesis, we conducted an experiment using the same apparatus and subjects as for the previous experiment (Section 5). We presented users with very short stimuli containing multiple circles with one circle flickering in some of the trials. Users

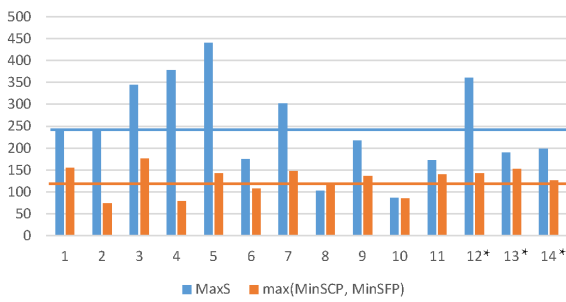


Figure 3: Size thresholds of circle diameters (in pixels) per participant (indicated by participant number) obtained for MaxS (blue) and the maximum of MinSCP and MinSFP (orange). The horizontal lines show the average values. (* Users 12 to 14 were tested with 144 Hz.)

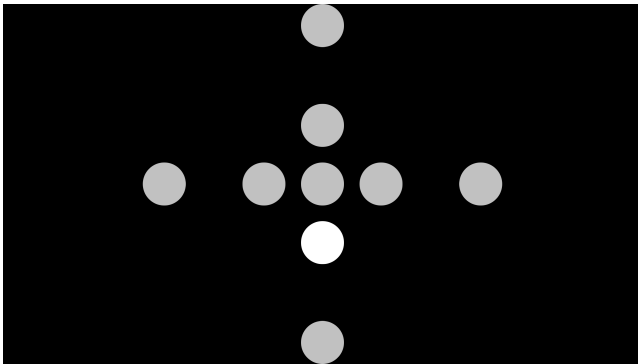


Figure 4: Example stimulus with nine circles. The target circle in the close periphery (bottom) is indicated by increased brightness for illustration purposes.

were asked to indicate whether they saw one of the circles flickering. We expected to see more correct responses for circles in the peripheral monitor regions than in the center of the monitor.

6.1. Stimuli

The stimuli consisted of nine circles placed in three rings: one circle was placed in the center of the monitor while the others were placed top, bottom, left or right of the center, in the near and far periphery, as in the MinSCP and MinSFP blocks in the previous experiment. Figure 4 shows an example stimulus.

The size of each circle was set to the user's maximum size threshold obtained in the MaxS-block of the preceding experiment (see Figure 2), or a maximum visual angle of 7.5° . The circles in the near and far periphery were placed as in the MinSCP and MinSFP blocks of the previous experiment (see Section 5). It is important to note that all circles had the same screen size in pixels. This means, that they were not equal in visual angle, which shrinks towards the periphery. The luminance was set according to the personally adjusted luminance values in the calibration step. The background was black as before.

6.2. Task and Procedure

In each trial, the users' task was to indicate whether they saw one of the circles flickering or not. Each stimulus was preceded by a fixation cross presented for two seconds on the center of the screen. After that, the stimulus (as shown in Figure 4) was presented for 0.75 seconds. Finally, a blank screen was shown, with a text asking the users to press a mouse button in case they detected a flickering circle, or the "f"-key if not. After giving the response, an inverse image (i.e., black circles on gray background) was shown for 0.1 seconds before showing the fixation cross again.

6.3. Design

For this experiment, we utilized a within-subjects design with four experimental conditions: *center*, *close periphery*, *far periphery*, and *target absent*, depending on if a one of the circles was flickering and at which eccentricity. In the target absent condition, there was no flickering target. The presentation sequence of the stimuli was randomized. The dependent variable was the correctness of response. For each condition, we aggregated the responses into a correctness ratio between 0 (no correct response) to 1 (all responses correct). Each condition was repeated nine times. In the close and far periphery condition, the location of the target (top, left, bottom, right) was chosen randomly.

6.4. Results

For each user, we determined the ratio of correctly answered questions per condition. Since the normality assumption was violated for this correctness ratio, we performed a non-parametric Friedman test (see also Figure 5). The test showed a significant difference in correctness depending on the test condition ($\chi^2(3) = 30.179, p < .001$). Bonferroni-corrected pairwise post-hoc comparisons for the three target-present conditions were performed using Wilcoxon-Signed Rank tests. We found significant differences between center and close periphery ($Z = -3.112, p = .002$) and between center and far periphery ($Z = -3.309, p = .001$). Also, there is a significant difference of correctness between close and far periphery ($Z = -2.668, p = .008$). On average, targets in the far periphery were the ones most correctly detected (95.2%; median: 100%), while targets in the center were missed in around 80% of all cases (21.4% correctness on average; median: 11).

These results confirm our hypothesis that high frequency flicker can more easily go unnoticed in the foveal than in the peripheral vision. While we could show that high frequency flicker in the periphery can be detected in almost 100% of all cases, our results suggest that high frequency flicker can sometimes be perceived even in the foveal vision. Mind, however, that we did not use an eye tracker and therefore cannot guarantee that users were indeed looking at the center of the screen (i.e., the center target) during the trials. The true positive responses for the center condition may be caused by fixations outside the center screen region. Also, improper calibration in the preceding experiment may have caused this effect – in particular for the single user scoring around 89% correct responses in the center condition (see outlier dot in the first column of Figure 5).

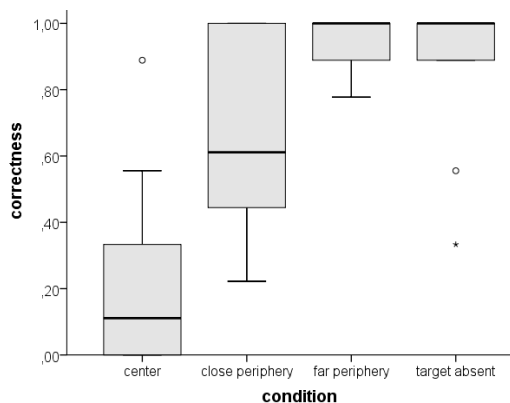


Figure 5: Box plots of correctness rate of responses for the four conditions (boxes for first to third quartile, whiskers up to 1.5 IQR, and outliers as individual dots).

7. Use Case: “Where’s Waldo”

High frequency flicker beyond the CFF for the foveal vision can be used to attract the user’s attention to image regions in the peripheral vision. To explore the effectiveness and user satisfaction with this technique, we invited the same 14 users to find hidden characters in a crowded scene. We used an image out of the series of “Where’s Waldo”, an illustrated children’s book showing colorful images of scenes crowded with numerous characters. Each image contains one instance of the main character Waldo, wearing a red-white striped pullover and hat, and some of his friends. The image we used was “The Gobbling Gluttons” from the book “Where’s Waldo: The Fantastic Journey.” It has been shown that unguided detection of “Waldo” can take up to several minutes [BOZ*14]. The flicker observer effect was used to sequentially guide the user’s attention to different characters in a single image.

As we used the findings from our psychophysics study (see Sections 5 and 6) to generate the flicker observer effect, we expected that users would easily spot the indicated characters, while expressing little discomfort caused by the flicker. User feedback was gathered quantitatively (through a questionnaire using a five-point Likert scale) and qualitatively through an unstructured feedback session. Our goal was not to keep the user’s gaze permanently fixated at the target, but to provide effective, yet unobtrusive support for target detection. Therefore, we let users report the location of the search target verbally instead of using an eye tracker.

7.1. Stimuli

To guide the users to the characters, we used the flicker observer effect: each character was surrounded by a 330 pixel region of interest, subject to high frequency flicker, of which 44 percent is affected by the dithering described below. The size of this region was the same for every user. The flicker effect was achieved by alternating between the original pixel colors and black. This effectively darkens the perceived brightness of the flickering patch (\rightarrow Talbot-Plateau Law). To compensate for this brightness difference, users had to adjust the brightness level of the remaining scene with respect to the flickering patch interactively at the beginning. Using

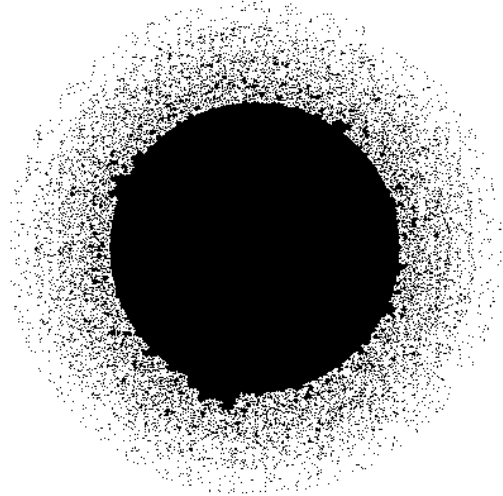


Figure 6: Image of dithering mask for seamlessly integrating high frequency flicker patches into images. The black pixels are flickering, the white pixels are not flickering.

the mouse scroll wheel, the CIE L*a*b* luminance component of all non-flickering pixels could be increased or decreased. The frequency of the flicker was the same as in the preceding experiments.

To avoid sharp boundaries between the flicker region and the context, we used a dithering mask for the flicker. The flickering area can be viewed as a circular region centered at the point c , with two radii, r_1 and r_2 , where $r_2 > r_1$. If the distance d between a point p and c is smaller than r_1 , then the pixel at p is flickering. If $r_2 > d > r_1$, then the probability that the pixel at p is flickering decreases linearly from 1 at r_1 to 0 at r_2 . We calculate this in a fragment shader by comparing the distance to a point in a texture with random values generated with Matlab. An example dithering mask can be seen in Figure 6.

7.2. Task and Procedure

The users were asked to find five different characters in the following order: The Wizard Whitebeard, Odlaw, Wenda, Waldo, and a Waldo Watcher. These characters were presented to them on a sheet of paper, as well as in a sidebar of the image (see Figure 7a). The area around the character that the user was currently looking for was flickering. When the users found a character, they had to click on it. The flickering area was then moved to the next character to be searched for. After the user found all the characters, a random section of the image was flickering, and the user was asked to describe different parts or characters in the image. Afterwards, the user was asked questions about how distracting the flickering area was, and whether he or she could detect it when looking directly at it.

7.3. Subjective User Feedback

As expected, all users could easily spot the characters indicated by the high frequency flicker within a few seconds. In comparison, earlier studies by Brown et al. [BOZ*14] show that without



Figure 7: The image used for the search task (“The Gobbling Gluttons” from the book “Where’s Waldo: The Fantastic Journey”). On the left, the positions of the targets have been marked with black circles. On the right a cutout contains the Wizard Whitebeard and a Waldo Watcher.

any attention guiding effect, users need around 500 seconds to find Waldo. All users reported that the flicker signal was clearly visible, even though the size of the flickering patch was fixed for all participants. This can be seen as an indication that an averagely sized flicker patch size can yield a satisfactory solution for a wide range of users. Most users also reported that high frequency flicker was distracting when attending to other image regions than the one currently flickering (average response for “It is easily possible to work when there is flicker in the periphery” was 3.64 on a five-point Likert scale). However, the discomfort generated by the high frequency flicker when directly looking at the image was rated as very low (with an average response for “The flicker was causing discomfort when looking at it” of 1.42). When asked specifically about it, most users reported they could not recall any discomfort or visual distortion associated with the image regions around the characters of interest.

According to this user feedback, high frequency flicker indeed seems to be a strong visual attractor – even in very cluttered scenes – while not having a strong influence on the perceived image quality. However, it can cause discomfort when users aim to direct their attention to other, non-flickering screen regions. Like other highlight techniques that have a strong effect on the surrounding image regions (like darkening [KMFK05], blurring [KMH02], or size compression [Fur86]), it is therefore not recommendable for longer exposure, or to indicate image regions of low degree of interest through high frequency flicker.

8. Applications of the Flicker Observer Effect

Since flicker does not strongly interfere with color or motion perception [POT08], it can be used for a variety of scenarios beyond our static use case presented above. For instance, high frequency flicker can direct the user’s gaze to the current ball location in a TV broadcast of a football match. It can draw the user’s attention to the notification area of a large display to inform the user about a recently arrived e-mail. It could also be employed for advertisement

purposes, so that the user is explicitly made aware of sponsored products placed in videos.

Previous research [Dav12] and our psychophysics experiments have shown that the flicker observer effect is restricted to a certain size range to work effectively. It is therefore necessary to define a region of interest (ROI) around the item to be highlighted that should be subject to high frequency flicker. To avoid sharp boundaries, we used dithering in our use case to smooth the transition between flickering and non-flickering regions. Each pixel in the ROI then has to be rendered bright and dark in alternating frames to create a flicker sensation, where the frame rate has to be twice as high as the CFF in the foveal region (around 60 Hz [TH93]). In practice, the higher the amplitude (i.e., the luminance difference between the bright and the dark color within the ROI), the stronger the effect [WLMB*14]. Thus, the technique works best for ROIs containing mostly pixels with medium brightness, which can be generated by alternating between very bright and very dark colors. The pixels in the ROI can be preprocessed to reduce the contrast of the ROI if the resulting luminance amplitude would be too low. This can be achieved by standard image contrast adjustment techniques.

To achieve a 60 Hz flicker, a display with a refresh rate of at least 120 Hz is necessary. Modern LCD and OLED TVs often provide 120 Hz or even 240 Hz refresh rate to reduce motion artifacts by interpolating between successive frames, while the normal TV signal is still 25 Hz (or 60 Hz in case of HDTV). Modern TV standards could add the possibility to specify a ROI for each TV frame that can be evaluated by suitable TV sets with sufficiently high refresh rate. The TV sets could then perform the required modifications of the ROI pixels and create the flickering sensation on-the-fly.

Computer monitors capable of active stereo rendering (i.e., producing stereoscopic 3D scenes with synchronized shutter glasses) could provide driver-based functionality to create high frequency flicker, as they are capable of high refresh-rates. The driver software could query operating system events like the mouse cursor position

or the pop-up of a notification window. It could then specify a ROI around the cursor to help the user keep track of the cursor on a large screen. It could make pop-up windows flicker to make them more evident for the user – without making them bigger (to avoid occlusion) or darkening the remaining display content (to avoid visual distortion of the remaining screen content).

9. Conclusions

In this paper, we have explored high frequency flicker as a means to effectively guide the user's attention in an image without noticeable changes to its visual appearance and without having to track the user's gaze. In a controlled experiment, we could demonstrate the *flicker observer effect*: on a consumer high frequency monitor, users are clearly aware of high frequency flicker in the peripheral vision, but rarely perceive the flicker when they are directly looking at it. We demonstrated the usefulness of this visual property by guiding users to hidden characters in a crowded scene using the flicker observer effect. With a fixed flicker patch size, users could easily find the characters, yet reported negligible discomfort or distraction caused by the high frequency flicker when looking at the image regions containing these characters, indicating that an average size flickering patch may be used for a wide range of users.

From our first experiment, it seems that a personal calibration routine is required to find the optimal flicker patch size and luminance offset. However, in the “Where’s Waldo” use case, we could show the effectiveness of the technique even for a fixed patch size. More in-depth experimentation is necessary to explore patch size and brightness settings that work across a wide range of users to support walk-up usage. We also plan to explore the interaction between flicker amplitude and patch size to unobtrusively highlight larger targets. Finally, it will also be interesting to formally investigate the effectiveness of the flicker observer effect in dynamic scenes, such as movies, games, or virtual environments.

Acknowledgements

This project has been funded by the Vienna Science and Technology Fund (WWTF) through project VRG11-010 and supported by EC Marie Curie Career Integration Grant through project PCIG13-GA-2013-618680 and the Austrian Science Fund FWF through project T 752-N30.

In Figure 7, the image “The Gobbling Gluttons” is Copyright ©1989 Martin Handford **From WHERE’S WALLY? THE FANTASTIC JOURNEY by Martin Handford**. Reproduced by permission of *Walker Books Ltd*, London SE11 5HJ

References

- [AHKMF11] ALPER B., HOLLERER T., KUCHERA-MORIN J., FORBES A.: Stereoscopic Highlighting: 2D Graph Visualization on Stereo Displays. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2325–2333. doi:10.1109/TVCG.2011.234. 2
- [BCP*09] BAUER F., CHEADLE S. W., PARTON A., MÜLLER H. J., USHER M.: Gamma flicker triggers attentional selection without awareness. *Proceedings of the National Academy of Sciences* 106, 5 (Feb. 2009), 1666–1671. URL: <http://www.pnas.org/content/106/5/1666>, doi:10.1073/pnas.0810496106. 3
- [BMC*12] BAILEY R., MCNAMARA A., COSTELLO A., SRIDHARAN S., GRIMM C.: Impact of subtle gaze direction on short-term spatial information recall. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (New York, NY, USA, 2012), ETRA '12, ACM, pp. 67–74. URL: <http://doi.acm.org/10.1145/2168556.2168567>, doi:10.1145/2168556.2168567. 2
- [BMSG09] BAILEY R., MCNAMARA A., SUDARSANAM N., GRIMM C.: Subtle Gaze Direction. *ACM Transactions on Graphics* 28, 4 (Sept. 2009), 1–22. URL: <http://doi.acm.org/10.1145/1559755.1559757>, doi:10.1145/1559755.1559757. 1, 2, 3
- [BOZ*14] BROWN E. T., OTTLEY A., ZHAO H., LIN Q., SOUVENIR R., ENDERT A., CHANG R.: Finding waldo: Learning about users from their interactions. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec 2014), 1663–1672. doi:10.1109/TVCG.2014.2346575. 7
- [BWC03] BARTRAM L., WARE C., CALVERT T.: Moticons: detection, distraction and task. *International Journal of Human-Computer Studies* 58, 5 (May 2003), 515–545. URL: <http://www.sciencedirect.com/science/article/pii/S1071581903000211>, doi:10.1016/S1071-5819(03)00021-1. 2
- [Dav55] DAVIS S. W.: Auditory and Visual Flicker-Fusion as Measures of Fatigue. *The American Journal of Psychology* 68, 4 (Dec. 1955), 654–657. URL: <http://www.jstor.org/stable/1418795>, doi:10.2307/1418795. 2, 5
- [Dav12] DAVSON H.: *Physiology of the Eye*. Elsevier, Dec. 2012. 1, 2, 8
- [Est10] ESTON P.: *Visual Perception A Clinical Orientation*, 3 ed. McGraw-Hill Companies, 2010, ch. 2, p. 16. 4
- [Eva92] EVANS B.: Visual display, Feb. 5 1992. EP Patent App. EP19,910,305,513. URL: <https://www.google.com/patents/EP0463787A3?cl=en>. 3
- [Far86] FARRELL J. E.: An analytical method for predicting perceived flicker. *Behaviour & Information Technology* 5, 4 (1986), 349–358. URL: <http://dx.doi.org/10.1080/01449298608914528>, arXiv:<http://dx.doi.org/10.1080/01449298608914528>, doi:10.1080/01449298608914528. 3, 4
- [Fuk79] FUKUDA T.: Relation between flicker fusion threshold and retinal positions. *Perceptual and Motor Skills* 49, 1 (1979), 3–17. URL: <http://dx.doi.org/10.2466/pms.1979.49.1.3>, arXiv:<http://dx.doi.org/10.2466/pms.1979.49.1.3>, doi:10.2466/pms.1979.49.1.3. 3
- [Fur86] FURNAS G. W.: Generalized fisheye views. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1986), ACM, pp. 16–23. doi:10.1145/22627.22342. 1, 2, 8
- [GBM07] GLUCK J., BUNT A., MCGRENERE J.: Matching Attentional Draw with Utility in Interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2007), ACM, pp. 41–50. doi:10.1145/1240624.1240631. 2
- [GF04] GUTWIN C., FEDAK C.: A Comparison of Fisheye Lenses for Interactive Layout Tasks. In *Proceedings of the 2004 Graphics Interface Conference* (2004), Canadian Human-Computer Communications Society, pp. 213–220. 1
- [HBW08] HOFFMANN R., BAUDISCH P., WELD D. S.: Evaluating visual cues for window switching on large screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), ACM, pp. 929–938. doi:10.1145/1357054.1357199. 2
- [HE12] HEALEY C. G., ENNS J.: Attention and Visual Memory in Visualization and Computer Graphics. *IEEE Transactions on Visualization and Computer Graphics* 18, 7 (2012), 1170–1188. doi:10.1109/TVCG.2011.127. 2
- [HV33] HECHT S., VERRIJP C. D.: Intermittent stimulation by light. *The Journal of General Physiology* 17, 2 (1933), 251–268. URL: <http://jgp.rupress.org/content/17/2/251>, arXiv:<http://jgp.rupress.org/content/17/2/251.full.pdf>, doi:10.1085/jgp.17.2.251. 3

- [KMF05] KHAN A., MATEJKA J., FITZMAURICE G., KURTENBACH G.: Spotlight: Directing Users' Attention on Large Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2005), ACM, pp. 791–798. doi:10.1145/1054972.1055082. 1, 2, 8
- [KMH02] KOSARA R., MIKSCH S., HAUSER H.: Focus+context taken literally. *IEEE Computer Graphics and Applications* 22, 1 (Jan. 2002), 22–29. doi:10.1109/38.974515. 1, 2, 8
- [KV06] KIM Y., VARSHNEY A.: Saliency-guided Enhancement for Volume Visualization. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (Sept. 2006), 925–932. doi:10.1109/TVCG.2006.174. 1, 2
- [Low03] LOWE R. K.: Animation and learning: selective processing of information in dynamic graphics. *Learning and Instruction* 13, 2 (Apr. 2003), 157–176. URL: <http://www.sciencedirect.com/science/article/pii/S095947520200018X>, doi:10.1016/S0959-4752(02)00018-X. 2
- [MB14] MATEESCU V. A., BAJIĆ I. V.: Can subliminal flicker guide attention in natural images? In *Proceedings of the 1st International Workshop on Perception Inspired Video Processing* (New York, NY, USA, 2014), PIVP '14, ACM, pp. 33–34. URL: <http://doi.acm.org/10.1145/2662996.2663012>, doi:10.1145/2662996.2663012. 3
- [MB16] MATEESCU V. A., BAJIĆ I. V.: Visual Attention Retargeting. *IEEE MultiMedia* 23, 1 (Jan. 2016), 82–91. doi:10.1109/MMUL.2015.59. 1, 2
- [MBS*12] MCNAMARA A., BOOTH T., SRIDHARAN S., CAFFEY S., GRIMM C., BAILEY R.: Directing Gaze in Narrative Art. In *Proceedings of the ACM Symposium on Applied Perception* (2012), SAP '12, ACM, pp. 63–70. URL: <http://doi.acm.org/10.1145/2338676.2338689>, doi:10.1145/2338676.2338689. 2
- [NB64] NELSON T. M., BARTLEY S. H.: The Talbot-plateau law and the brightness of restricted numbers of photic repetitions at CFF. *Vision Research* 4, 7–8 (Oct. 1964), 403–411. URL: <http://www.sciencedirect.com/science/article/pii/0042698964900124>, doi:10.1016/0042-6989(64)90012-4. 3
- [POT08] PINTO Y., OLIVERS C. N. L., THEEUWES J.: Selecting from dynamic environments: Attention distinguishes between blinking and moving. *Perception & Psychophysics* 70, 1 (Jan. 2008), 166–178. URL: <http://link.springer.com/article/10.3758/PP.70.1.166>, doi:10.3758/PP.70.1.166. 2, 8
- [Rob11] ROBINSON A. C.: Highlighting in Geovisualization. *Cartography and Geographic Information Science* 38, 4 (Jan. 2011), 373–383. doi:10.1559/15230406384373. 2
- [RR88] ROVAMO J., RANINEN A.: Critical flicker frequency as a function of stimulus area and luminance at various eccentricities in human cone vision: A revision of granit-harper and ferry-porter laws. *Vision Research* 28, 7 (1988), 785 – 790. URL: <http://www.sciencedirect.com/science/article/pii/0042698988900259>, doi:http://dx.doi.org/10.1016/0042-6989(88)90025-9. 2
- [Sid97] SIDEBOTTOM S.: *Effects of Illumination and Viewing Angle on the Modeling of Flicker Perception in CRT Displays*. PhD thesis, Virginia Polytechnic Institute and State University, 1997. URL: <http://scholar.lib.vt.edu/theses/available/etd-4229173039731191/unrestricted/flicker.pdf>. 3, 4
- [SNHW05] SEITZ A. R., NANEZ J. E., HOLLOWAY S. R., WATANABE T.: Visual experience can substantially alter critical flicker fusion thresholds. *Human Psychopharmacology: Clinical and Experimental* 20, 1 (Jan. 2005), 55–60. URL: <http://onlinelibrary.wiley.com/doi/10.1002/hup.661/abstract>, doi:10.1002/hup.661. 2, 5
- [SOK*16] STROBELT H., OELKE D., KWON B. C., SCHRECK T., PFISTER H.: Guidelines for Effective Usage of Text Highlighting Techniques. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (Jan. 2016), 489–498. doi:10.1109/TVCG.2015.2467759. 2
- [Tal34] TALBOT H.: XLIV. experiments on light. *Philosophical Magazine Series 3* 5, 29 (1834), 321–334. URL: <http://dx.doi.org/10.1080/14786443408648474>, arXiv: <http://dx.doi.org/10.1080/14786443408648474>, doi:10.1080/14786443408648474. 2
- [TH90] TYLER C. W., HAMER R. D.: Analysis of visual modulation sensitivity. iv. validity of the ferry-porter law. *J. Opt. Soc. Am. A* 7, 4 (Apr 1990), 743–758. URL: <http://josaa.osa.org/abstract.cfm?URI=josaa-7-4-743>, doi:10.1364/JOSAA.7.000743. 3
- [TH93] TYLER C. W., HAMER R. D.: Eccentricity and the Ferry-Porter law. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision* 10, 9 (Sept. 1993), 2084–2087. 3, 8
- [Tyl85] TYLER C. W.: Analysis of visual modulation sensitivity. ii. peripheral retina and the role of photoreceptor dimensions. *J. Opt. Soc. Am. A* 2, 3 (Mar 1985), 393–398. URL: <http://josaa.osa.org/abstract.cfm?URI=josaa-2-3-393>, doi:10.1364/JOSAA.2.000393. 3
- [Tyl87] TYLER C. W.: Analysis of visual modulation sensitivity. iii. meridional variations in peripheral flicker sensitivity. *J. Opt. Soc. Am. A* 4, 8 (Aug 1987), 1612–1619. URL: <http://josaa.osa.org/abstract.cfm?URI=josaa-4-8-1612>, doi:10.1364/JOSAA.4.001612. 1
- [VMFS11] VEAS E. E., MENDEZ E., FEINER S. K., SCHMALSTIEG D.: Directing attention and influencing memory with visual saliency modulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), ACM, pp. 1471–1480. doi:10.1145/1978942.1979158. 1, 2
- [War12] WARE C.: *Information Visualization: Perception for Design*. Elsevier, May 2012. 2
- [WB04] WARE C., BOBROW R.: Motion to Support Rapid Interactive Queries on Node-link Diagrams. *ACM Transactions on Applied Perception* 1, 1 (July 2004), 3–18. doi:10.1145/1008722.1008724. 2
- [WLMB*14] WALDNER M., LE MUZIC M., BERNHARD M., PURGATHOFER W., VIOLA I.: Attractive Flicker – Guiding Attention in Dynamic Narrative Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec. 2014), 2456–2465. doi:10.1109/TVCG.2014.2346352. 2, 8
- [ZWSK97] ZHAI S., WRIGHT J., SELKER T., KELIN S.-A.: Graphical Means of Directing Users' Attention in the Visual Interface. In *Human-Computer Interaction INTERACT '97*. Springer US, 1997, pp. 59–66. 1, 2

List of Figures

1.1	Example of the dual search task. There is one green square, surrounded by red squares and green circles.	1
2.1	Figure 12-12 from [Sch09] by Schwarz. This figure illustrates how the retina of the eye is constructed. The different types of cells and how they are connected are shown. These connections and functions affect how light is interpreted, and need to be taken into consideration when creating a visualization. . .	6
2.2	Figure 10.2 from The Nature and Status of Visual Resources [Fra13] by Franconeri. This image shows the competition for attention in the visual system. Here, the thicker black lines are connected to the players. Because they are the focus of attention, the visual stream regarding the gorilla is not payed any attention, a phenomenon called “inattentional blindness”. We use this blindness to our advantage in Paper B.	7
2.3	Figure 4 from “Analysis of visual modulation sensitivy. III. Meridional variations in peripheral flicker sensitivity” by Tyler [Tyl87]. The images shows measured critical flicker frequency sensitivities for two subjects. The results show that flicker sensitivity is higher in the near periphery (approx 15°-45°) than in foveal or far peripheral vision. This is used in Paper D.	16
3.1	Warping of color map according to different test subjects and measurements	22
3.2	Chameleon (Paper B) coloring method for different levels of the HIV data set.	23
3.3	Diagram shows how the forces work. Red arrow indicates origin, black arrows indicate forces between objects and toward origin (parent). Here each circle indicates a group of children.	24
3.4	Placement of nodes on the hue circle. Parents are on outer circle, children are on inner. Color indicates group.	26
3.5	Application of Cuttlefish on a zoomable treemap [Gan16]. The two images show data regarding population size and geographical location. The larger the rectangle, the larger the population. The color circles below the images show the assigned colors.	26
3.6	Testing and results from the Flicker Observer Effect paper (see Paper D). The top image shows the placement of circles which were used to measure the sensitivity to flicker. The bottom boxplots show the results.	28
		79

Bibliography

- [AGFC12] Israel Abramov, James Gordon, Olga Feldman, and Alla Chavarga. Sex and vision II: color appearance of monochromatic lights. *Biology of Sex Differences*, 3(1):21, Sep 2012.
- [BCG⁺07] Rigmor C. Baraas, Joseph Carroll, Karen L. Gunther, Mina Chung, David R. Williams, David H. Foster, and Maureen Neitz. Adaptive optics retinal imaging reveals S-cone dystrophy in tritan color-vision deficiency. *J. Opt. Soc. Am. A*, 24(5):1438–1447, May 2007.
- [BCS07] Luciana Bernardi, Vital P. Costa, and Lineu Oto Shiroma. Flicker perimetry in healthy subjects: influence of age and gender, learning effect and short-term fluctuation. *Arquivos Brasileiros de Oftalmologia*, 70:91 – 99, 02 2007.
- [BK09] David Bimler and John Kirkland. Colour-space Distortion in Women who are Heterozygous for Colour Deficiency. *Vision Research*, 49(5):536 – 543, 2009.
- [BKJ04] David L. Bimler, John Kirkland, and Kimberly A. Jameson. Quantifying variations in personal color spaces: Are there sex differences in color vision? *Color Research & Application*, 29(2):128–134, 2004.
- [BMB14] A. E. Boehm, D. I. A. MacLeod, and J. M. Bosten. Compensation for red-green contrast loss in anomalous trichromats. *Journal of Vision*, 14(13):19, 2014.
- [BMC⁺12] Reynold Bailey, Ann McNamara, Aaron Costello, Srinivas Sridharan, and Cindy Grimm. Impact of Subtle Gaze Direction on Short-term Spatial Information Recall. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 67–74, New York, NY, USA, 2012. ACM.
- [Boy92] Robert M. Boynton. *Human Color Vision*. Optical Society of America, 1992.

- [BRJM05] J.M. Bosten, J.D. Robinson, G. Jordan, and J.D. Mollon. Multidimensional scaling reveals a color dimension unique to ‘color-deficient’ observers. *Current Biology*, 15(23):R950 – R952, 2005.
- [BRT95] Lawrence D Bergman, Bernice E Rogowitz, and Lloyd A Treinish. A Rule-based Tool for Assisting Colormap Selection. In *Proceedings of the 6th conference on Visualization’95*, page 118. IEEE Computer Society, 1995.
- [BSM⁺15] Jürgen Bernard, Martin Steiger, Sebastian Mittelstädt, Simon Thum, Daniel Keim, and Jörn Kohlhammer. A Survey and Task-Based Quality Assessment of Static 2d Colormaps. In *Proc. SPIE*, volume 9397, pages 9397–9397–16, 2015.
- [BWC03] Lyn Bartram, Colin Ware, and Tom Calvert. Moticons: Detection, Distraction and Task. *International Journal of Human-Computer Studies*, 58(5):515–545, May 2003.
- [BWMG08] Jibin Bao, Yuanyuan Wang, Yu Ma, and Xiaodong Gu. Re-coloring Images for Dichromats Based on an Improved Adaptive Mapping Algorithm. In *2008 International Conference on Audio, Language and Image Processing*, pages 152–156, July 2008.
- [CCB11] W. Chen, W. Chen, and H. Bao. An Efficient Direct Volume Rendering Approach for Dichromats. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2144–2152, Dec 2011.
- [CFM⁺13] R. Carnecky, R. Fuchs, S. Mehl, Y. Jang, and R. Peikert. Smart Transparency for Illustrative Visualization of Complex Flow Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):838–851, May 2013.
- [Che73] Herman Chernoff. The Use of Faces to Represent Points in k-Dimensional Space Graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.
- [Dac96] D M Dacey. Circuitry for color coding in the primate retina. *Proceedings of the National Academy of Sciences*, 93(2):582–588, 1996.
- [Dav95] S. W. Davis. Auditory and Visual Flicker-Fusion as Measures of Fatigue. *The American Journal of Psychology*, 68(4):654–657, 1995.
- [Dav12] Hugh Davson. *Physiology of the Eye*. Elsevier, December 2012.
- [DR90] R. F. Dacheux and E. Raviola. Physiology of HI Horizontal Cells in the Primate Retina. *Proceedings of the Royal Society of London B: Biological Sciences*, 239(1295):213–230, 1990.

- [EF98] Fritz Ebner and Mark D. Fairchild. Development and Testing of a Color Space (IPT) with Improved Hue Uniformity. *Color and Imaging Conference*, 1998(1):8–13, 1998.
- [Eva92] B.T. Evans. Visual display, February 5 1992. EP Patent App. EP19,910,305,513.
- [Far86] JOYCE E. Farrell. An analytical method for predicting perceived flicker. *Behaviour & Information Technology*, 5(4):349–358, 1986.
- [FG12] David R. Flatla and Carl Gutwin. Situation-Specific Models of Color Differentiation. *ACM Trans. Access. Comput.*, 4(3):13:1–13:44, December 2012.
- [F.R34] H.F. Talbot Esq. M.P. F.R.S. Xliv. experiments on light. *Philosophical Magazine*, 5(29):321–334, 1834.
- [Fra13] Steven L. Franconeri. The Nature and Status of Visual Resources. In Daniel Reisberg, editor, *The Oxford Handbook of Cognitive Psychology (Oxford Library of Psychology)*, chapter 10, pages 147–162. Oxford University Press, Oxford, 2013.
- [Fry92] Glenn A. Fry. Confusion lines of dichromats. *Color Research & Application*, 17(6):379–383, 1992.
- [Fuk79] Tadahiko Fukuda. Relation between Fucker Fusion Threshold and Retinal Positions. *Perceptual and Motor Skills*, 49(1):3–17, 1979.
- [Gan16] Ganeshv. Zoomable Treemap Template. <http://bl.ocks.org/ganeshv/6a8e9ada3ab7f2d88022>, 2016. [Online; accessed 13-December-2016].
- [GBM07] Jennifer Gluck, Andrea Bunt, and Joanna McGrenere. Matching Attentional Draw with Utility in Interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 41–50. ACM, 2007.
- [GH30] R. Granit and P. Harper. Comparative studies on the peripheral and central retina: Synaptic reactions in the eye. ii. *American Journal of Physiology*, 95:941–945, 1930.
- [GOTG05] Amy A. Gooch, Sven C. Olsen, Jack Tumblin, and Bruce Gooch. Color2Gray: Saliency-preserving Color Removal. In *ACM SIGGRAPH 2005 Papers, SIGGRAPH '05*, pages 634–639, New York, NY, USA, 2005. ACM.
- [Gre10] Donna L. Gresh. Self-Corrected Perceptual Colormaps. <http://researchweb.watson.ibm.com/people/g/donnagresh/colormaps.pdf>, 2010. [Online; accessed 24-August-2017].

- [Gre15] Ernest Greene. Evaluating letter recognition, flicker fusion, and the talbot-plateau law using microsecond-duration flashes. In Denis G. Pelli, editor, *PloS One*, volume 10. Public Library of Science, 2015.
- [HB03] Mark Harrower and Cynthia A Brewer. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal*, 40(1):27–37, 2003.
- [HBW08] Raphael Hoffmann, Patrick Baudisch, and Daniel S. Weld. Evaluating Visual Cues for Window Switching on Large Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 929–938. ACM, 2008.
- [HCJW09] J. B. Huang, C. S. Chen, T. C. Jen, and S. J. Wang. Image recolorization for the colorblind. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1161–1164, April 2009.
- [HCN⁺05] Heidi Hofer, Joseph Carroll, Jay Neitz, Maureen Neitz, and David R. Williams. Organization of the Human Trichromatic Cone Mosaic. *Journal of Neuroscience*, 25(42):9669–9679, 2005.
- [Hea96] Christopher G Healey. Choosing Effective Colours for Data Visualization. In *Visualization'96. Proceedings.*, pages 263–270. IEEE, 1996.
- [HMPJ06] S.M. Hood, J.D. Mollon, L. Purves, and G. Jordan. Color Discrimination in Carriers of Color Deficiency. *Vision Research*, 46(18):2894 – 2900, 2006.
- [HV33] Selig Hecht and Cornelis D. Verrijp. Intermittent stimulation by light. *The Journal of General Physiology*, 17(2):251–268, 1933.
- [Iha03] Ross Ihaka. Colour for Presentation Graphics. In Kurt Hornik, Friedrich Leisch, and Achim Zeileis, editors, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, 2003.
- [JDBM10] Gabriele Jordan, Samir S. Deeb, Jenny M. Bosten, and J. D. Mollon. The dimensionality of color vision in carriers of anomalous trichromacy. *Journal of Vision*, 10(8):12, 2010.
- [JKW⁺12] J. Y. Jeong, H. J. Kim, T. S. Wang, S. W. Jung, and S. J. Ko. Real-time Video Re-Coloring Algorithm Considering the Temporal Color Consistency for the Color-Blind. *IEEE Transactions on Consumer Electronics*, 58(2):721–729, May 2012.
- [KKL⁺15] Barbora Kozlikova, Michael Krone, Norbert Lindow, Martin Falk, Marc Baaden, Daniel Baum, Ivan Viola, Julius Parulek, and Hans-Christian Hege. Visualization of Biomolecular Structures: State of the Art. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015.

- [KOF08] 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, Nov 2008.
- [KRC02a] G. Kindlmann, E. Reinhard, and S. Creem. Face-based Luminance Matching for Perceptual Colormap Generation. In *IEEE Visualization, 2002. VIS 2002.*, pages 299–306, Nov 2002.
- [KRC02b] Gordon Kindlmann, Erik Reinhard, and Sarah Creem. Face-based Luminance Matching for Perceptual Colormap Generation. In *Proceedings of the Conference on Visualization '02, VIS '02*, pages 299–306, Washington, DC, USA, 2002. IEEE Computer Society.
- [KSH01] T. Kohonen, M. R. Schroeder, and T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001.
- [Kue01] Rolf G. Kuehni. Determination of unique hues using Munsell color chips. *Color Research & Application*, 26(1):61–66, 2001.
- [KV06] Y. Kim and A. Varshney. Saliency-guided Enhancement for Volume Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):925–932, Sept 2006.
- [LFK⁺13] Sharon Lin, Julie Fortuna, Chinmay Kulkarni, Maureen Stone, and Jeffrey Heer. Selecting Semantically-resonant Colors for Data Visualization. In *Proceedings of the 15th Eurographics Conference on Visualization, EuroVis '13*, pages 401–410. The Eurographs Association; John Wiley; Sons, Ltd., 2013.
- [LH92] H. Levkowitz and G. T. Herman. Color scales for image data. *IEEE Computer Graphics and Applications*, 12(1):72–80, Jan 1992.
- [LMAPV15] Mathieu Le Muzic, Ludovic Autin, Julius Parulek, and Ivan Viola. celVIEW: A Tool for Illustrative and Multi-scale Rendering of Large Biomolecular Datasets. In *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine, VCBM '15*, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2015. Eurographics Association.
- [LSS13] S. Lee, M. Sips, and H. P. Seidel. Perceptually Driven Visibility Optimization for Categorical Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1746–1757, Oct 2013.
- [LWY⁺09] Bo Liu, Meng Wang, Linjun Yang, Xiuqing Wu, and Xian-Sheng Hua. Efficient Image and Video Re-coloring for Colorblindness. In *2009 IEEE International Conference on Multimedia and Expo*, pages 906–909, June 2009.

- [Mar98] Paul R. Martin. Colour processing in the primate retina: recent progress. *The Journal of Physiology*, 513(3):631–638, 1998.
- [MB79] Donald I. A. MacLeod and Robert M. Boynton. Chromaticity Diagram Showing Cone Excitation by Stimuli of equal Luminance. *J. Opt. Soc. Am.*, 69(8):1183–1186, Aug 1979.
- [MB14] Victor A. Mateescu and Ivan V. Bajić. Can Subliminal Flicker Guide Attention in Natural Images? In *Proceedings of the 1st International Workshop on Perception Inspired Video Processing, PIVP '14*, pages 33–34, New York, NY, USA, 2014. ACM.
- [MB16] V. A. Mateescu and I. V. Bajic. Visual Attention Retargeting. *IEEE MultiMedia*, 23(1):82–91, Jan 2016.
- [MBS⁺12] Ann McNamara, Thomas Booth, Srinivas Sridharan, Stephen Caffey, Cindy Grimm, and Reynold Bailey. Directing Gaze in Narrative Art. In *Proceedings of the ACM Symposium on Applied Perception, SAP '12*, pages 63–70. ACM, 2012.
- [MBS⁺14] Sebastian Mittelstädt, Jürgen Bernard, Tobias Schreck, Martin Steiger, Jörn Kohlhammer, and Daniel A. Keim. Revisiting Perceptually Optimized Color Mapping for High-Dimensional Data Analysis. In *EuroVis 2014 : the Eurographics Conference on Visualization ; 9-13 June 2014, Swansea, Wales, UK ; EuroVis Short Papers*, pages 91–95. Eurographics Association, 2014.
- [MD85] J Moran and R Desimone. Selective attention gates visual processing in the extrastriate cortex. *Science*, 229(4715):782–784, 1985.
- [Med16] Medialab. i want hue. <http://tools.medialab.sciences-po.fr/iwanthue/>, 2016. [Online accessed 24-August-2017].
- [MO10] Gustavo M. Machado and Manuel M. Oliveira. Real-Time Temporal-Coherent Color Contrast Enhancement for Dichromats. *Computer Graphics Forum*, 29(3):933–942, June 2010. Proceedings of EuroVis.
- [MPMP12] Ian J. Murray, Neil R. A. Parry, Declan J. McKeefry, and Athanasios Panorgias. Sex-related differences in peripheral human color vision: A color matching study. *Journal of Vision*, 12(1):18, 2012.
- [NN11] Jay Neitz and Maureen Neitz. The genetics of normal and defective color vision. *Vision Research*, 51(7):633 – 651, 2011. Vision Research 50th Anniversary Issue: Part 1.
- [OKMK14] H. Orii, H. Kawano, H. Maeda, and T. Kouda. Color Conversion Algorithm for Color Blindness Using Self-organizing Map. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS)*

and 15th International Symposium on Advanced Intelligent Systems (ISIS), pages 910–913, Dec 2014.

- [PCH11] J. Park, J. Choi, and D. Han. Applying Enhanced Confusion Line Color Transform Using Color Segmentation for Mobile Applications. In *2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, pages 40–44, May 2011.
- [PG88] Ronald Pickett and Georges Grinstein. Iconographic Displays For Visualizing Multidimensional Data. In *Proc. IEEE Conf. Syst. Man Cybern.*, volume 1, pages 514–519, 09 1988.
- [PTCS06] Dorothe A. Poggel, Bernhard Treutwein, Claudia Calmanti, and Hans Strasburger. Increasing the temporal gain: Double-pulse resolution is affected by the size of the attention focus. *Vision Research*, 46(18):2998 – 3008, 2006.
- [RGW05] Karl Rasche, Robert Geist, and James Westall. Re-coloring Images for Gamuts of Lower Dimension. *Computer Graphics Forum*, 24(3):423–432, 2005.
- [Ric77] F M Richards. Areas, Volumes, Packing, and Protein Structure. *Annual Review of Biophysics and Bioengineering*, 6(1):151–176, 1977. PMID: 326146.
- [Ric81] Jane S. Richardson. The Anatomy and Taxonomy of Protein Structure. *Advances in Protein Chemistry*, 34:167 – 339, 1981.
- [SBM⁺15] Martin Steiger, Jürgen Bernard, Sebastian Mittelstädt, Marco Hutter, Daniel Keim, Simon Thum, and Jörn Kohlhammer. Explorative Analysis of 2D Color Maps. In *Proceedings of WSCG*, pages 151–160, 2015.
- [SC99] Daniel J Simons and Christopher F Chabris. Gorillas in Our Midst: Sustained Inattentional Blindness for Dynamic Events. *Perception*, 28(9):1059–1074, 1999. PMID: 10694957.
- [Sch09] Steven Schwartz. *Visual Perception A Clinical Orientation Fourth Edition*. McGraw-Hill Education / Medical, 2009.
- [Sid97] Shane Sidebottom. *Effects of Illumination and Viewing Angle on the Modeling of Flicker Perception in CRT Displays*. PhD thesis, Virginia Polytechnic Institute and State University, 1997.
- [SL98] Daniel J. Simons and Daniel T. Levin. Failure to detect changes to people during a real-world interaction. *Psychonomic Bulletin & Review*, 5(4):644–649, Dec 1998.

- [SMO⁺13] B. Sajadi, A. Majumder, M. M. Oliveira, R. G. Schneider, and R. Raskar. Using Patterns to Encode Color Information for Dichromats. *IEEE Transactions on Visualization and Computer Graphics*, 19(1):118–129, Jan 2013.
- [SNHW05] Aaron R. Seitz, Jose E. Nanez, Steven R. Holloway, and Takeo Watanabe. Visual experience can substantially alter critical flicker fusion thresholds. *Human Psychopharmacology: Clinical and Experimental*, 20(1):55–60, 2005.
- [SP98] Andrew Stockman and Daniel J Plummer. Color from invisible flicker: a failure of the talbot–plateau law caused by an early ‘hard’ saturating nonlinearity used to partition the human short-wave cone pathway. *Vision Research*, 38(23):3703 – 3728, 1998.
- [TdJ14] M. Tennekes and E. de Jonge. Tree Colors: Color Schemes for Tree-Structured Data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2072–2081, Dec 2014.
- [TH90] Christopher W. Tyler and Russell D. Hamer. Analysis of visual modulation sensitivity. IV. Validity of the Ferry–Porter law. *J. Opt. Soc. Am. A*, 7(4):743–758, Apr 1990.
- [TH93] C. W. Tyler and R. D. Hamer. Eccentricity and the Ferry–Porter law. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, 10(9):2084–2087, September 1993.
- [TKM⁺14] H. Takagi, H. Kudo, T. Matsumoto, N. Ohnishi, and Y. Takeuchi. Measurement of Confusion Color Pairs for Dichromats in Order to Use Applications Supporting Color Vision Deficiency. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 3458–3464, July 2014.
- [Tyl87] Christopher W. Tyler. Analysis of visual modulation sensitivity. iii. meridional variations in peripheral flicker sensitivity. *J. Opt. Soc. Am. A*, 4(8):1612–1619, Aug 1987.
- [TZ16] G. Tennenholtz and I. Zachevsky. Natural Contrast Enhancement for Dichromats Using Similarity Maps. In *2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*, pages 1–5, Nov 2016.
- [VBM99] Françoise Viénot, Hans Brettel, and John D. Mollon. Digital video colourmaps for checking the legibility of displays by dichromats. *Color Research & Application*, 24(4):243–252, 1999.
- [VMFS11] Eduardo E. Veas, Erick Mendez, Steven K. Feiner, and Dieter Schmalstieg. Directing Attention and Influencing Memory with Visual Saliency Modulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’11*, pages 1471–1480, New York, NY, USA, 2011. ACM.

- [VS17] John E. Vanston and Lars Strother. Sex Differences in the Human Visual System. *Journal of Neuroscience Research*, 95(1-2):617–625, 2017.
- [War88] C. Ware. Color Sequences for Univariate Maps: Theory, Experiments and Principles. *IEEE Computer Graphics and Applications*, 8(5):41–49, Sept 1988.
- [WB04] Colin Ware and Robert Bobrow. Motion to Support Rapid Interactive Queries on Node–link Diagrams. *ACM Transactions on Applied Perception*, 1(1):3–18, July 2004.
- [WGM⁺08] Lujin Wang, Joachim Giesen, Kevin T McDonnell, Peter Zolliker, and Klaus Mueller. Color Design for Illustrative Visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1739–1754, 2008.
- [WLMB⁺14] M. Waldner, M. Le Muzic, M. Bernhard, W. Purgathofer, and I. Viola. Attractive Flicker – Guiding Attention in Dynamic Narrative Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2456–2465, December 2014.
- [ZH16] L. Zhou and C. D. Hansen. A Survey of Colormaps in Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(8):2051–2069, Aug 2016.