

# Color Design for Illustrative Visualization

Lujin Wang, Joachim Giesen, Kevin T. McDonnell, *Member, IEEE*, Peter Zolliker, *Member, IEEE*, and Klaus Mueller, *Senior Member, IEEE*

**Abstract**—Professional designers and artists are quite cognizant of the rules that guide the design of effective color palettes, from both aesthetic and attention-guiding points of view. In the field of visualization, however, the use of systematic rules embracing these aspects has received less attention. The situation is further complicated by the fact that visualization often uses semi-transparencies to reveal occluded objects, in which case the resulting color mixing effects add additional constraints to the choice of the color palette. Color design forms a crucial part in visual aesthetics. Thus, the consideration of these issues can be of great value in the emerging field of illustrative visualization. We describe a knowledge-based system that captures established color design rules into a comprehensive interactive framework, aimed to aid users in the selection of colors for scene objects and incorporating individual preferences, importance functions, and overall scene composition. Our framework also offers new knowledge and solutions for the mixing, ordering and choice of colors in the rendering of semi-transparent layers and surfaces. All design rules are evaluated via user studies, for which we extend the method of conjoint analysis to task-based testing scenarios. Our framework's use of principles rooted in color design with application for the illustration of features in pre-classified data distinguishes it from existing systems which target the exploration of continuous-range density data via perceptual color maps.

**Index Terms**—Color design, volume rendering, transparency, user study evaluation, conjoint analysis, illustrative visualization

## 1 INTRODUCTION

Recent years have seen multifarious efforts to better integrate and exploit properties of human visual perception into visualization design. Illustrative rendering techniques have been developed that render the scene at different levels of abstractions [30] and detail [32] or in different rendering styles [5], with applications ranging from information visualization [20] to full-scale volume rendering. In these approaches, the levels of abstraction are most often controlled by a task- or object-dependent importance parameter [31]. Another perception-motivated strategy is to guide viewer attention to salient features [16]. Color can play a major role in these particular efforts. However, there is no illustrative rendering system so far that incorporates rules from color design directly into the visualization engine. Yet, the working scenario of graphics designers is quite similar to that of illustrative rendering. Like in graphics design, pre-classified or segmented objects typically constitute the working set. It thus seems beneficial to learn from the well-developed rules used by graphics designers, working in highly profit-oriented fields such as advertising, and adapt and modify these rules for the special needs of illustrative data visualization.

The study of color for data visualization tasks is not new, yet it has traditionally focused on the design of transfer functions (also called color scales or color palettes) intended for the mapping of scientific or cartographic scalar data to RGB attributes. Well known for the former is the PRAVDA system [2], and for the latter the Color Brewer [4], which has been recently generalized in [35].

We address a different data scenario – one in which a volume or image has been divided into a set of coherent regions, e.g., specific organs in a volumetric medical dataset, cells in a microscopic image, or graphical objects in an information display. Further, any of these regions may bear interesting intensity variations which need to be visually represented in a faithful manner.

- Lujin Wang and Klaus Mueller are with the Center for Visual Computing, Stony Brook University, E-Mail: {lujin, mueller}@cs.sunysb.edu
- Kevin McDonnell is with Dowling College, E-Mail: mcdonnek@dowling.edu
- Joachim Giesen is with the Friedrich-Schiller-Universität Jena, E-Mail: giesen@minet.uni-jena.de
- Peter Zolliker is with EMPA Dübendorf, E-mail: peter.zolliker@empa.ch

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.  
For information on obtaining reprints of this article, please send e-mail to: [tcvg@computer.org](mailto:tcvg@computer.org).

Professional designers and artists are quite cognizant of rules that guide the design of color palettes, not only from an aesthetic point of view but also from an attention-guiding, salient one. Likewise, visualization is not only concerned with providing a pleasing image – it also has a mission, that is, to help users gain quick and accurate insight into the visualized data [34]. Our framework captures these known color design rules into a knowledge-based system, which then combines them with scene analysis, user preferences, and importance functions to derive appropriate colorizations. These latter considerations are typically not embodied in currently existing frameworks.

The scope of our system is image-centric as well as volume visualization. Both often use semi-transparencies to reveal occluded objects. Here, color mixing effects and the perceived depth-ordering of the semi-transparent layers pose additional constraints on the color choices. Since illustrative rendering is not always physically correct, proper user studies are crucial for the parameterization of the models underlying our system. We are particularly interested in the effectiveness of a given colorization to solve certain perceptual tasks and to derive knowledge that is then applicable for practical visualization tasks. For this, we extended the method of *conjoint analysis* [11] from choice-based testing to task-based testing.

We note that in this paper we only consider the effects of color, and not those of illustrative style and the combination of these. We believe that a decoupling of these visualization parameters is necessary to develop a rudimentary framework, which can then later be applied in the context of stylistically more advanced systems.

Our paper is structured as follows. Section 2 presents related work, Section 3 gives a system overview, and Section 4 provides theoretical considerations. Section 5 focuses on the color design aspects, and Section 6 discusses the color mixing. Section 7 extends conjoint analysis to task-based testing, applies it to derive models for color mixing in an illustrative rendering context, and demonstrates their use in a specific application. Section 8 ends with conclusions.

## 2 RELATED WORK

Good visual aesthetics make a visualization task an enjoyable one and therefore reduces stress, which is a significant factor in today's fast paced data analysis scenarios. As [25] suggests, it is essential that products designed for use under stress follow good human-centered (pleasant, aesthetic) design, since stress makes people less able to cope with difficulties, less flexible, and less creative in problem solving tasks. In fact, it has been shown that objects considered

beautiful stimulate different areas in the brain than those considered unattractive [14]. The role of color as a means to increase image aesthetics has been studied for a long time in the arts and design literature. The landmark texts by Itten [13] and Wong [36] provide great insight on the human perception of color and its aesthetic aspects. Much information is also available in the books by Stone [29] and Ware [34]. One popular design aspect is color harmony, which is a fairly old concept, and a quantitative representation was described by [23]. It consists of three perceptual coordinates: hue, value (lightness/brightness), and chroma (colorfulness). In search of an intuitive 2D representation for visual designers, Itten then arranged the harmonic colors into a color wheel, which reduced (flattened) this color space to mostly variations in hue. Later, [18] employed Itten's wheel, in conjunction with extensive psycho-physical studies, to introduce a set of 80 harmonic colors schemes. This system was used in the automated image color harmonization framework of [8].

Another important use of color is to focus attention [1], known as *pop-out*. Pop-out exploits the low-level mechanisms of the visual system, which enable humans to detect visual properties very rapidly (although they may not so do consciously). Pop-out is strongly related to the vividness of a color patch, as well as its size and the degree to which it differs from the vividness of other colors in the scene [7] (and local brightness contrast).

Color is generally used to label graphical objects. But how many distinct colors (labels) are reasonable? User studies [12] explored colored target identification capabilities and found that users did quite well when the number of colors was 5 or less. These deficiencies are partially rooted in the limitations of human working memory. Also limiting the maximal numbers of colors is the phenomenon of simultaneous contrast which changes the appearance of a color based on its surround [34]. It widens the required "safety margin" of a given color and limits the number of these. Pop-out is attractive since no extra colors (hues) are required to generate attention effects.

In [27] it was demonstrated that color maps should preserve a monotonic mapping in luminance, and [17] suggested that the best mapping results from a straight line through a perceptual color space such as CIE  $L^*a^*b^*$  (also called CIELAB or, for short, Lab color space). All of these approaches involve all three perceptual components of color: hue, saturation, and brightness. Our system also makes use of all three color components, but in a more intuitive yet free-form fashion, where users only pick the most intuitive component, the hue, and the system optimizes the other two.

Relevant is also the interactive color palette tool proposed by [21], designed to support creative interactions for graphics designers. This user group typically demands ample artistic freedom for producing their artwork. But, in contrast to computational scene rendering typically used in visualization, graphics designers tend to only produce a few sheets of artwork a day. Our system replaces the ad-hoc, manual scene analysis of graphics designers by computational scene analysis, followed by automated rule-based parameter settings.

Finally, there has been a substantial amount of work on color transparency [9] in the perception literature, with the goal to define accurate models for the prediction of perceived transparency. Physical models were shown to be best suited for this task [10], which use subtractive color mixing (spectral filtering). These types of spectral models are also widely employed in computer graphics rendering of photo-realistic participating media [19]. On the other hand, the goal of visualization, and in particular illustrative visualization, is not necessarily physical realism, but rather effective information depiction, and user studies must be conducted for tuning these illustrations to ensure proper information communication. In previous work [11] we have shown that conjoint analysis is an effective method for this.

### 3 SYSTEM OVERVIEW

Inherent to our system is a feature-based approach. It assumes that the data have been segmented or grouped into coherent structures, which may be spatially distributed and only logically assigned to classes. This and further information, such as importance or uncer-

tainty, is then used in the semi-interactive colorization procedure that is the subject of this paper. Thus, it is classification followed by colorization, not classification via a continuous value-color mapping with color scales and transfer functions, which has been amply addressed in the past. In our visualizations, pixels of the same density might receive different colors if they belong to different classified structures. The structures presented for colorization could even originate from a fusion of a multi-modal dataset. Thus, our framework applies to any pre-classified rendering pipeline, such as 2-level volume rendering [15] and nearly all illustrative rendering systems.

Our system is geared towards a wide range of visualization researchers and practitioners who would like to inject their personal color preferences into their visualization designs or rendering engines, but who are not experts in graphic design or do not have the time for lengthy parameter tweaking. First, the user is free to choose the hue of the features in the scene, bringing to bear personal preferences, while the system assists by making suggestions on aesthetic rules, such as color harmony. We chose Itten's color wheel paradigm as the user interface, since it is convenient and intuitive. Following this manual hue specification, the system then calculates and optimizes the harder-to-choose parameters (saturation, luminance), applying its encoded color design rules in conjunction with provided importance and computed scene parameters, such as feature size, density ranges, and interactions. All of these calculations are done in a perceptual color space, we use CIELAB.

As is well known, contrast is important for distinguishing both local and global features. Our system provides two layers of contrast, in an organized and methodical way. It uses hue for inter-class contrast, vividness to tag class importance (for highlighting), and lightness to provide intra-class contrast. Further, it also captures the possibly important local density dynamics of the data in the lightness channel. In scanned and simulated datasets the scalar densities usually map to certain material properties, and it is desirable to preserve these variations as lightness modulations.

As mentioned, both volume and information visualization often use semi-transparencies to reveal occluded objects, and the color mixing that can occur when two or more overlapping semi-transparent surfaces are composited can result in radically different colors. This can be disturbing, distracting, and also confusing, especially if this mixed color has already been reserved for another scene object. We propose a set of rules designed to avoid undesired color mixing. Further, in semi-transparent volume rendering there are potential ambiguities with respect to visual depth-ordering, that is, detecting which object is in front and which in the back. We also propose a set of rules that resolve and avoid these occlusion ambiguities. The rules formulated in our system are then tested in a set of targeted user studies, based on the conjoint analysis framework described in [11], which we formally extend here to task-based testing.

### 4 THEORETICAL CONSIDERATIONS

As mentioned, our aim is to combine known color design rules with scene analysis at various levels, user preferences, and importance functions to colorize scalar data in information displays. For this, we seek to incorporate various principles of vision psychology, such as low-level visual processing, emotion, and aesthetics, in order to better control the specific visualization task at hand. In cognitive science, color is a psychological experience (that is, we can also imagine it) of three orthogonal components: quality (the hue), quantity (the lightness, which is perceived brightness), and purity (the chroma or saturation). To conceive our automated color design system, we can take advantage of a well established body of knowledge in color design and perceptual science. Next, we briefly enumerate the set of rules relevant for our goals. These rules are given in addition to those of color harmony, which we extend in a companion paper [33].

**R1:** Vivid colors (bright, saturated colors) stand out. They guide attention to a particular feature, generating the pop-out effect.

**R2:** An excessive amount of vivid colors is perceived as unpleasant and overwhelming; use them between duller background tones.

**R3:** Foreground-background separation works best if the foreground color is bright and highly saturated, while the background is de-saturated.

**R4:** Colors can be better discriminated if they differ simultaneously in hue, saturation and lightness.

**R5:** The low end lightness steps should be very small, while the high end requires larger steps (Weber's Law).

**R6:** Discrimination is poorer for small objects. Hue, saturation and lightness discrimination all decrease.

**R7:** Complementary (opponent) colors are located opposite on the color wheel and have the highest chromatic contrast. When mixing opponent colors they may cancel each other, giving neutral grey.

**R8:** Some hues appear inherently more saturated than others. Yellow has the least number of perceived saturation steps (10). For hues on both sides of yellow, the saturation steps increase linearly.

**R9:** An opposite effect of R8 is that the brightest lights fall in the yellow range, while blues, violets (purples) and reds are least bright.

**R10:** For labeling, apart from black, white, grey, there are 4 primary colors (red, green, blue, yellow) and 4 secondary colors (brown, orange, purple, pink). Also, the number of color labels should be  $\leq 6-7$ .

**R11:** Warm colors (red, orange, yellow) excite emotions, grab attention. Cold colors (green to violet) create openness and distance.

**R12:** Important for hue-based labeling is the fact that increasing the lightness (and saturation) does not change the perceived hue.

**R13:** Also important for labeling is that objects of similar hue are perceived as a group, while objects of different hues are perceived as belonging to different groupings.

We note that R10 refers to the *naming* of colors, as in human language. This is different from the color hierarchy generated by additive color *mixing* where the primary colors (red, green, blue) generate the secondary colors (yellow, cyan, magenta) and mixing a primary with a secondary color generates a tertiary color. Also, brown as well as pink are special colors. Brown is only perceived amid brighter color contrasts, while pink is in fact a de-saturated red.

An implication of rule R8 is that viewers will find small differences in saturation for blue, violet and red highly discriminable, while small differences in yellow saturation will be hard to detect. Green and orange are in the middle. R13 implies that color can be used to organize the display into perceptual chunks, viewed pre-attentively. The color layout indicates that parts of the image form distinct areas, and if the same color appears in different parts of the image, these areas appear linked together, suggesting that they have something in common. On the other hand, variations in a basic color can convey variations within the class data. Similar colors, adjacent on the color wheel, suggest a similarity relationship, while different colors, located on more opposite sides of the color wheel, suggest a difference relationship between objects and areas of the screen.

We have incorporated these rules and guidelines into the following three steps of our color section framework. Section 5 will then outline in detail how this framework is used in image generation:

**Step 1** (assisted hue selection via color wheel for inter-class contrast): R3 (for background selection), R7 (for semi-transparent rendering), R8 and R9 (when highlighting), R10, R11, R13.

**Step 2** (automatic vividness selection for class and class component highlighting): R1, R2, R6, R8.

**Step 3** (automatic lightness selection for intra-class contrast): R4, R5, R12.

## 5 COLOR SELECTION FRAMEWORK

The user selects the hue via the color wheel popularized by Itten and also used by [8], which is a composite of the conventions for color naming and mixing. It consists of twelve hues, including red, orange, yellow, green, cyan, blue, purple, and magenta, arranged over the hue channel of the HSV color space (hence the name *HSV color wheel*). We define similar hues as either being in the same named category (R10) or located adjacently on this HSV color wheel (R13). After hue selection, the system calculates vividness and lightness in CIELAB (Lab) color space. Since at this stage we only have a HSV color, this requires a number of conversion operations, which are described next. A reverse procedure then produces the RGB triple for rendering. In the following, we shall use the HSV space for the illustration of all relationships since it is the most intuitive space when hue is the fixed variable. We also could have used the CIE LCH(ab) space [6] which separates lightness, chroma, and hue, but we chose HSV since this is the color space of our user interface.

### 5.1 Conversion algorithms for vividness and lightness

In order to calculate the lightness  $L$  and vividness  $V$  of a HSV color, it is first transferred to sRGB space, then to CIE XYZ space, and finally to Lab color space (see Figure 1 to the left). The value range of  $L$  is  $[0, 100]$ , with 0 for black and 100 for white. We define  $V$  as a color's relative purity (or chroma). The absolute chroma of a color can be computed in Lab space as follows ( $L$  is orthogonal to  $a, b$ ):

$$\text{chroma}(h, s, v) = \sqrt{a \cdot a + b \cdot b} \quad (1)$$

Although the maximum chroma values for different hues are quite different, the value range for vividness is always  $[0, 1]$ :

$$\text{vividness}(h, s, v) = \text{chroma}(h, s, v) / \text{chroma}(h, 1, 1) \quad (2)$$

There are various non-linear relationships when transforming an  $(h, s, v)$  triple to Lab space ( $L, V$ ) and back. The conversion to  $(r, g, b)$  is only piecewise linear, and the conversion to Lab space is strictly non-linear. Consequently the colors of equi-lightness (and equi-vividness) in HSV space reside on non-linear trajectories. Figure 2a shows an *equi-lightness curve*, which is comprised of all colors from A to B that have the same  $L$  on this hue slice. Likewise, Figure 2b shows all colors with the same  $V$  on this hue slice, giving rise to the *equi-vividness curve*, which is also the *equi-chroma curve*.

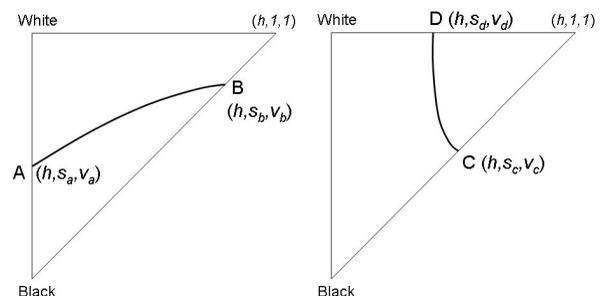


Fig. 2: Illustration of equi-lightness and equi-vividness curves in HSV color space: (a) an equi-lightness curve, (b) an equi-vividness curve.

The monotonic relationships enable efficient binary search procedures for the mapping. Figure 3 shows examples of equi-lightness curves on two different HSV hue slices. Some curves have been labeled with their lightness value. From the bottom curve to the top curve, the lightness increases linearly. For different hues, the lightness values of the most vivid colors (the top-most outside points on the hue slices) are quite different. We observe that vivid cyan has a much higher lightness than vivid red (this can also be observed in the Munsell color tree [24]). Next, Figure 4 shows examples of equi-vividness curves on these hue slices. Some curves have been labeled with their chroma values. Note that the chroma interval between adjacent curves is the same. From the right-most curve to the left-most curve, the vividness decreases from 1 to 0. We see that the

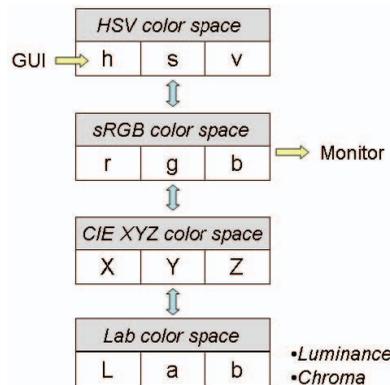


Fig. 1: Color spaces in our system.

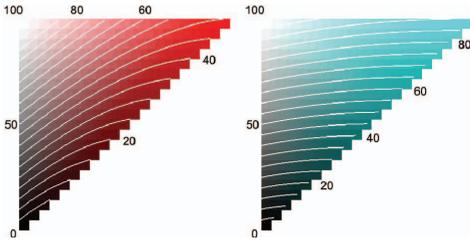


Fig. 3: Examples of equi-lightness curves in HSV color space. (a) Hue slice with  $h = 0$ , (b) Hue slice with  $h = 180$ .

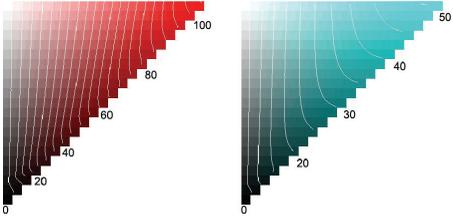


Fig. 4: Examples of equi-vividness curves in HSV color space. (a) Hue slice with  $h = 0$ , (b) Hue slice with  $h = 180$ .

absolute chroma varies substantially for the different hues. Cyan has considerably fewer distinguishable saturation steps than red (see R8).

To obtain the  $s$  and  $v$  values from a given  $h, L, V$  we intersect, on this  $h$  slice, the equi-lightness curve for  $L$  (Figure 2a) with the equi-vividness curve for  $V$  (Figure 2b) to obtain the color with the (desired)  $L$  and  $V$ . However, no two equi-lightness curves and equi-vividness curves will intersect – in some cases it is impossible to obtain a color with both the desired lightness and the desired vividness. To compensate, we have to either sacrifice  $L$  or  $V$ . In order to obtain the desired lightness, the vividness of the color may have to be sacrificed, or vice versa. Since lightness contrast is crucial to help distinguish features on the local level, we prefer to keep lightness, or avoid this problem by optimization in the selection procedure.

## 5.2 Color selection algorithm

As mentioned, the data should be available in segmented form. The smallest unit is defined as *Object*, and several objects with the same properties belong to one *Class*. Each object has attributes, such as its area, and each class has an importance value. A class computes its total area from its member objects. Each class may have a user-specified importance value. Currently, our framework supports three importance levels: most (3), less (2), and least (1) important, but this can be easily extended. The color selection process is as follows:

**Step 1** (Hue selection): Starting from the most important class, for each class, the hue wheel is activated with the proper hue categories from which users may select the class hue. The system suggests hues of warm colors for the most important classes and hues of cool or neutral colors for the least important classes (R11). For objects or classes with high importance but small area our system suggests hues where high vividness is associated with high lightness, such as cyan (Figure 3 and 4). All objects in one class will be assigned the same main hue. A harmonic color template [8] provides guidelines on harmonic hue selections.

**Step 2** (Vividness selection): Next, the vividness of each class is computed based on class importance and class area. For the former, we may assign a vividness of 1 for the highest importance, 0.7 for less, and 0.3 for the least. The relative area (class area vs. image size) is then used to adjust the vividness in order to avoid a large area to receive a very high vividness, and a global weighting parameter can be tuned to account for the real image size. Thus, important classes will be colored with a higher vividness, and for the same importance level, a class with smaller area will also be colored with a higher vividness, while a class with larger area (and the same importance) will receive a color with lower vividness. Importance variations among the objects within a class can also be brought out, by varying

their vividness. So, if there is a large range of importance variations within a class, our system will suggest a hue (in step 1) that supports this granularity, such as red or blue (recall Figure 4 and R8).

**Step 3** (Lightness selection): Good global lightness contrast is very important to help discriminate different features (R4, R12). Further, sufficient local lightness contrast is also required to visually convey small feature detail. Our lightness selection framework achieves both. We first compute the lightness for the most important classes, according to their hue and vividness, and then assign the less important classes lightness values a pre-defined distance away from the lightness values already used (see Figure 5). We only choose values within a user-specified range  $[L_{low}, L_{high}]$ , which determines the overall lightness of the visualization. For each less or least important class, our algorithm starts from  $(L_{high} + L_{low})/2$  and then uses a randomized algorithm [22] for generating lightness values at both the lighter and darker sides to find a lightness which has a pre-defined minimal distance  $L_{dis}$  from all lightness values used so far. This lightness value generator makes use of the equi-lightness curves (Figure 3) to ensure that the assigned vividness can be maintained. We initialize  $L_{dis}$  to a certain value, for example 10, and then gradually decrease this value if the selection process does not succeed after a few iterations. Once the hue, lightness, and vividness have been selected, we convert these  $(h, L, V)$  triples to their  $(h, s, v)$  equivalents and finally to RGB for display.

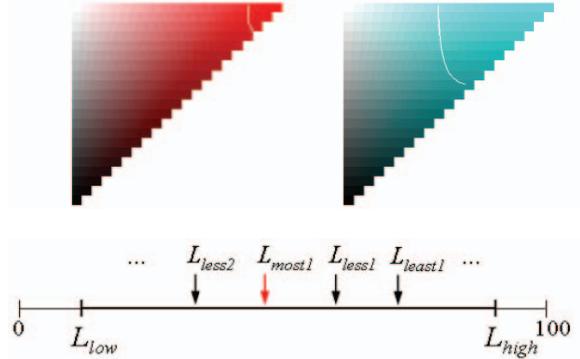


Fig. 5: Illustration of the lightness selection. The class in red with highest importance has higher vividness, and the class in cyan with less importance has lower vividness.  $L_{most1}$  is the lightness selected for the class in red. All other classes (including the one colored in cyan) will receive the other lightness levels shown. The lightness range levels,  $L_{low}$  and  $L_{high}$ ,

Figure 6 illustrates our color design algorithm using a 2D colorization of an image of biological cells, generated via Transmission Electron Microscopy (TEM). This colorization also demonstrates how the original image grey-level intensities modulate the lightness of the assigned object colors (here the cells). In this figure, the labels denote the object classes, colored according to their respective importance.

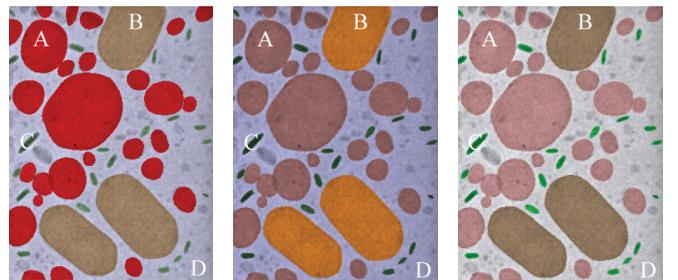


Fig. 6: Colorizations of Transmission Electron Microscopy (TEM) data. From left to right, classes A, B, or C (the small oblong, elliptical cells) were most important.

## 6 COLOR MIXING FRAMEWORK

Let us assume two semi-transparent overlapping objects (layers), that is, object 1 in front and object 2 in the back, with colors ( $C_1, C_2$ ) and opacities ( $\alpha_1, \alpha_2$ ). Then, using front-to-back color compositing, the mixed color  $C = C_1\alpha_1 + C_2\alpha_2(1 - \alpha_1)$ . Thus the weights for the two colors are given as  $W_1 = \alpha_1$ , and  $W_2 = \alpha_2(1 - \alpha_1)$ . This mixed color  $C$  will have higher lightness than either  $C_1W_1$  or  $C_2W_2$ . If the opacity values for two objects are the same, then the front color contributes more than the back color. Figure 7 shows two color mixing examples. The opacity is 0.4 for all objects. We observe that new hues between red and green are generated in Figure 7a (red circle in front gives orange) and 7c (green circle in front gives yellow). The color wheels to the left of the figures explain that when red is more heavily weighted than green, the average color is orange, while in the opposite case the average color is yellow. On the other hand, as indicated by Figure 7b and 7d, when two opposite hues are mixed, no false colors are generated, and the mixed color is either neutral or with a hint of the front object's hue. Thus, this color choice helps prevent potential problems resulting from color mislabeling.

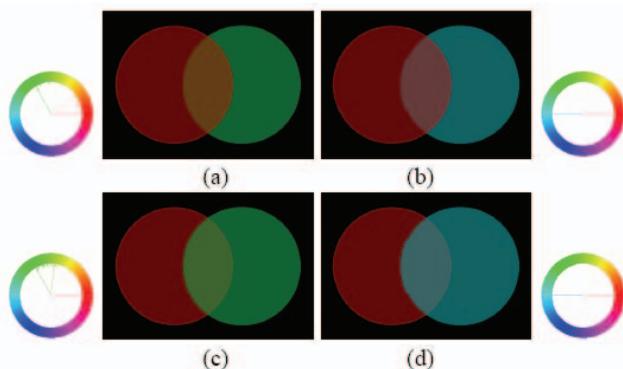


Fig. 7. Two color mixing examples. (a) and (c) Red and green are mixed with red in front or green in front. (b) and (d) Red and cyan are mixed with red in front or cyan in front.

Figure 8 demonstrates the color mixing problem for semitransparent volume renderings. There we see that when cyan and yellow are mixed, green hues will be generated. In contrast, a blue and yellow opposite-color combination will keep the original hues.

The opposite-color method just described will generate neutral colors, with possibly just a hint of the front color, in the overlap regions. However, sometimes it is important to reveal the color of the back (interior) object, especially when it is fully enclosed. We accomplish this by decreasing the saturation of the exterior (outside) object. This reveals the interior object's real color, without having to change the alpha channel of the transfer function. As Figure 9 shows, the color of the inside feature is disclosed increasingly more as the color of the outside feature becomes less saturated. The center setting appears to represent a good compromise, still keeping some of the outside object's hue, while allowing the interior object's hue (the yellowish skeleton) to show through with minimal color distortion.

When two or even more features are embedded, the color of an interior feature will always be occluded or changed. In this case, at least one hue category will fall between the two others on the hue wheel and a new hue will necessarily be generated. To minimize these adverse effects, we propose to assign two opposite hues for the two most important objects and more neutral colors for the lesser important object. To be safe, however, we should avoid using any hue which can be generated by mixing two hues already chosen. Figure 10a shows a three-color mixing example, with three objects A, B, and C. We observe that by reducing the vividness (saturation) of object C, while keeping its lightness, we can reduce the false color in the B, C overlap region.

Sometimes other constraints dictate the choice of certain colors, preventing the selection of opposite hues for partially overlapping objects. To provide a solution even in these cases, we devised what

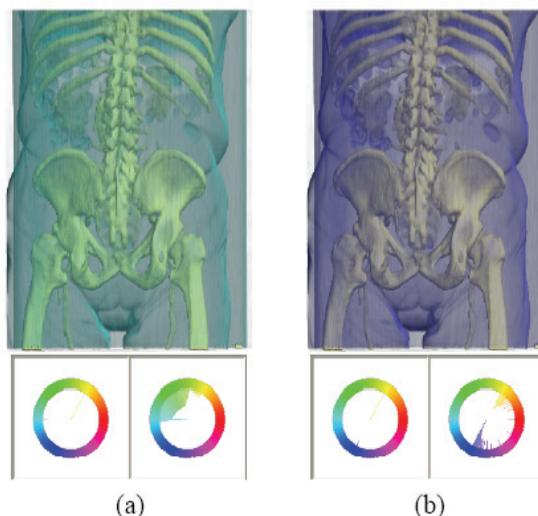


Fig. 8. Color mixing in a volume rendered body. (a) Cyan and yellow are mixed, (b) Blue and yellow are mixed, Corresponding hue wheels are placed below each rendering, the left hue wheel is for transfer function, and the right is for the rendering.

we refer to as *local solution*, aimed at reducing false colors in the overlap regions. Our scheme is demonstrated in Figure 10b. It works by reducing the saturation of the color in the rear object only in the overlap region  $A \cap B$ , while keeping its lightness (we can detect such regions efficiently with a depth-peeling algorithm). After the color mixing, this region clearly has more hint of the front-object color. In fact, this local solution is very general – it can also be used to reduce the false-color effect in areas in which more than two objects overlap (an example is given later in this paper, in Figure 12).

Considering again Figure 7, we also observe that the ordering of objects can not be visually preserved all the time. While Figures 7c and 7d show the correct ordering without any doubt, Figures 7a and 7b are less convincing. We hypothesized that unsaturated colors will not preserve the ordering very well, and we also hypothesized that vivid colors with high lightness should be chosen for the front object. However, the relationship between colors and perceived orderings brings complications. For example, if the user already assigned red in front of cyan, either increasing the lightness of red, which will reduce its saturation, or decreasing the lightness of cyan, which makes it look more transparent, will not help provide the correct ordering. Only increasing the opacity (weight) of the front object will yield the correct ordering. This is shown in Figure 10c, where the opacities of red and cyan are 0.6 and 0.4 respectively. However, at the same time, in Figure 10b, we observe that our local solution also seems to visually preserve the correct ordering. Therefore, in order to obtain more insight into these phenomena and parameterize our various solutions, we conducted a thorough user study. For this we extended the method of conjoint analysis from comparative to task-based testing. Both are discussed next, in Section 7.

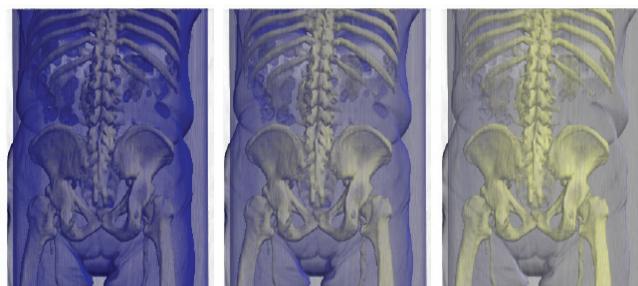


Fig 9: Revealing the color of an inside object by decreasing the saturation of outside objects. Left to right: the yellow hue shows when the blue becomes less saturated.

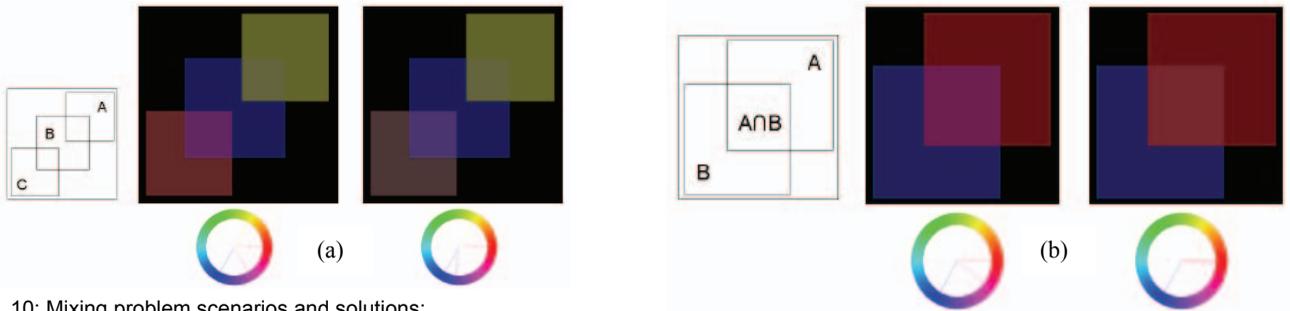


Fig. 10: Mixing problem scenarios and solutions:

(a) Three-color mixing examples. (left) Object labels, (center) A and B are assigned opposite hues: yellow and blue, C is red and thus the overlapping region of B and C gives a false color, which can be observed from the hue wheel as well. (right) The saturation of C is reduced, but its lightness is kept, which reduces the false color.

(b) Local solution to reduce false color. (left) Object labels, A is in front of B. (center) A false color is generated when mixing red and blue. (right) The false color is reduced by our local solution.

(c) Ordering-preserving color mixing example, using a higher weight for the object in front. Red and cyan are mixed – red is in front.

## 7 TEST AND APPLICATION OF THE COLOR MIXING FRAMEWORK

We conducted our user study with a fairly simple experimental setup. The participants, all graphics-educated users, were presented a sequence of 24 images showing two colored, overlapping disks on a LCD computer screen with black background (such as the one in Figure 10c). For all disk pairs we had an intended depth order. The participants were asked to choose for each image the disk they perceived to be in front by clicking with a mouse on a label next to the disks. The participants were tested for color deficiencies via a standard Ishihara test, and data collected from color-deficient participants were discarded from our analysis. We had 72 persons without color deficiencies participating, providing a total of 1728 data points.

Each task, i.e., choosing in an image the disk perceived to be in front, provides a binary answer, namely, ‘correct’ (ground truth) if the choice coincides with the intended ordering, or ‘incorrect’ otherwise. This is a conjoint measurement with respect to the ground truth in the sense that not just one, but many parameters can jointly contribute to the outcome. Statistically isolating the effect of each of these parameters is one of the main strengths of conjoint analysis.

Thus far, conjoint analysis has been applied mainly to measure preferences, where the raw data are assessed in choice experiments. Participants in such studies are asked to choose the preferred object from a selection of a few objects, which are drawn from a larger collection of objects. In our previous work [11] these ‘objects’ were visualizations (images). In this current work, we aim for a different goal, namely, we seek to test for the ability of a given visualization to elicit, in a user, the desired cognitive response. More concretely, a user’s correct (incorrect) choice of the front disk can be directly attributed to the success (failure) of the employed (multi-parametric) visualization method chosen to convey this relation. We therefore devised an extension of conjoint analysis from choice-based to this new task-based setting, where we observe binary (correct/incorrect) labels on a single image instead of preferences among different images. Choice data and task data cannot be analyzed in exactly the same way, which requires adapting the statistical framework presented in [11] to task-based conjoint analysis data.

### 7.1 Studied visualization parameters and their analysis

In our analysis, we studied the following visualization parameters:

**Foreground Color:** The hue of the disk intended to be in the foreground. This parameter has the values red (Red), yellow (Yel), green (Grn), cyan (Cyan), blue (Blu) and magenta (Mag).

**Background Color:** The hue of the disk intended to be in the back. This parameter also has the values red (Red), yellow (Yel), green (Grn), cyan (Cyan), blue (Blu) and magenta (Mag). No images with identical foreground and background colors were used.

**Mixing Weights:** These specify the  $\alpha$ -values for front ( $\alpha_f$ ) and back ( $\alpha_b$ ) disk when compositing the colors of the overlapping disks. We used the standard compositing equation. The four levels ( $w_0, w_1, w_2, w_3$ ) were  $(\alpha_f, \alpha_b) = (0.5, 0.4), (0.4, 0.4), (0.4, 0.667), (0.4, 0.75)$ . We note that the third-level net compositing weights are identical.

**Saturation/Lightness:** The vividness  $V$  is always 1, which determines the full lightness that can be obtained. In the following, ‘avg  $L$ ’ is the average lightness of the disk intended to be in front and the one in the back. The four levels are: (LV0, LV1, LV1, LV3) = (Full  $L$  for both disks), (Full  $L$  for front, avg  $L$  for back disk), (avg  $L$  for front, full  $L$  for back disk), (avg  $L$  for both disks).

**Position:** Describes where the disk intended to be in front is presented. Possibilities are on left (Left) or right (Right) of the image.

**Algorithm:** Two algorithms were tested for color mixing. Hence, the algorithm parameter has two levels: (Global) and (Local). We also analyzed our data by either considering only images rendered using the local color mixing algorithm or only images using global mixing. Thus, in these cases we only analyzed parts of the data.

Each image is uniquely identified by the parameter levels (or values) set for foreground / background color, mixing weight, saturation / lightness, position, and algorithm. As mentioned, we observe, from each user the binary values ‘correct’ or ‘incorrect’ for each presented image. In each such image, all parameters, or better, their set levels, jointly contribute to the success or failure in the user’s task of detecting the disk in front – some more and some less. Our goal is to quantify each parameter level’s contribution to this outcome, that is, the foreground/background separation. This contribution is called the *part-worth* of the parameter level. The variance of the part-worths of one parameter is a measure for this parameter’s importance. When this variance is large, then we can improve the quality of an image significantly (in terms of our objective) by changing from a parameter value with low part-worth to one with high part-worth. Thus, parameters with large variance are more promising for design improvement, as it matters (much) more to move from an unimportant to an important level (value) than for parameters with small variance.

We adapted the statistical model for choice-based conjoint analysis [11] to analyze also task-based conjoint analysis data. The model assumes that the value of an image can be computed as the sum of the part-worth values of its parameter levels. We statistically verified the validity of this linearity assumption using  $\chi^2$  tests (see [11]). The tests supported this assumption for all pairs of parameters, except when we used all data in the analysis, that is, all images rendered using either local or global color mixing. In the latter case the combination of the parameter ‘algorithm’ with the parameter ‘background color’ has a significant non-linear behavior. Figure 11 summarizes the calculated part-worths along with their estimated standard errors (see [11]) for all parameters and levels listed above.

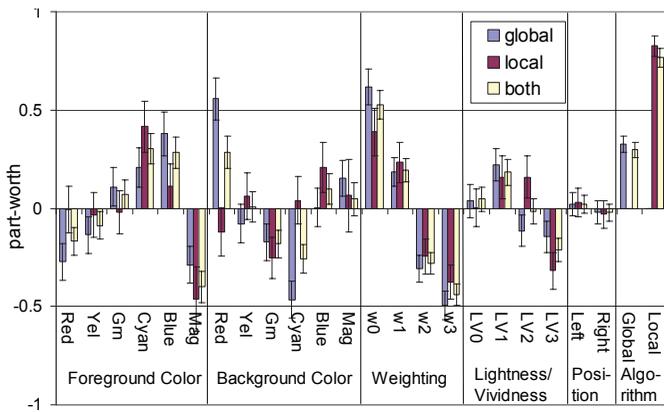


Fig. 11. Part-worths for all parameters and all their levels. The associated error (bracketed bars) is expressed in standard deviations. Shown are the results of three analyses: only global algorithm images (rendered via global mixing), only local algorithm images, and all images (using both local and global algorithm images).

We found the relative importance (variance) of the parameters to be ranked as follows: (1) weighting: 0.323; (2) global-local: 0.231; (3) foreground color: 0.186, (4) background color: 0.131, L/V: 0.109, position (L/R): 0.02. Hence, the most profitable parameters are weight and local/global enhancement, followed by the remaining parameters. Position was found not to be a factor, which was expected.

To interpret our conjoint analysis study, a disk intended to be in front is easier to identify if it has been rendered with a parameter value that has a large positive assigned part-worth, and it is more difficult to identify if a parameter value with an assigned large negative part-worth is present. Following our statistical model [11] the value for an image is the sum of the normally distributed part-worths of the parameter levels that were used when rendering the image. The sum of normally distributed random variables is itself normally distributed. Using the cumulative distribution function of a normal distribution we can compute, from the value of an image, the probability that a test person (from a population similar to the one from which we collected our data) will correctly identify the disk intended to be in front. Obviously this success probability is greater for larger values. In that sense, the value of an image is a measure for the probability that a test person will correctly identify the disk in front. The part-worth value for the local (global) level of the ‘algorithm’ parameter in the local (global) analysis is exactly the offset needed to recover the observed success probability. In the analysis that considers all images this offset is added to both the local and the global level of the ‘algorithm’ parameter. The results in Figure 11 indicate:

**Algorithm:** We observe that our local mixing strategy significantly aids in correctly identifying the disk intended to be in front (in fact, this parameter is marked by the greatest positive part-worth).

**Mixing weights:** Weighting the color of the disk intended to be in front more heavily (that is, increasing its opacity) helps to identify

the ordering, whereas more weight on the color of the disk in the back is counter to this ability. This can be expected from prior work.

**Background / foreground color:** Our analysis indicates a trend that cold colors, such as green, blue or cyan in front, and warm colors, such as red, yellow or magenta in the back enhance foreground/background separation. Blue tends to be seen in front even if its opacity is low and the mixed color is basically that of the disk in the back. It was expected that warm/cold or cold/warm colors had to be used to encode the front/back relation. But we unexpectedly observed that cold/warm (for front/back) is a much more effective than warm/cold. The latter relationship would have been explained by chromostereopsis (lens refraction causes red objects to be perceived closer than blue objects), but we note that chromostereopsis is generally stated for objects viewed side by side, but not overlapping and mixed in color. We therefore believe that the color mixing has a decisive effect on this unexpected study outcome and is probably the key to explain the deviation from what was expected. Another interesting observation is that the part-worth values for foreground and background color parameters sort the colors around a color wheel (most pronounced for the foreground color in the global analysis). In fact, we consider these findings as the most interesting of our study, and we plan to analyze them in more detail in subsequent research.

**Saturation/lightness:** We find that increasing the lightness for front-objects and decreasing the lightness for back-objects also helps in the depth-ordering task.

**Position:** We do not observe a presentation bias (which is expected), that is, it is irrelevant if the disk intended to be in front is shown on the left or on the right. This can also be noted from the small variance (importance) of the position parameter.

**7.2 Practical application: Illustrative Parallel Coordinates**

We incorporated the rules and parameter settings derived in this user study into our knowledge base. We now demonstrate a practical application of our system in the context of our illustrative parallel coordinate system [20]. These examples also show that our findings are directly applicable to situations where objects of three different colors overlap one another. Figure 12 shows several visualizations of just six dimensions of the eight-dimensional, 1,356-point scan bio dataset (courtesy of <http://davis.wpi.edu/~xmdv>). The curved bands illustrate 6-dimensional clusters obtained by a k-means clustering of the dataset. Figure 12a presents these clusters with the assignment of a highly-weighted, cool (blue) color to the cluster in front. It is easy to perceive that the blue-colored cluster, with an opacity of 0.7, is in front of the red cluster ( $\alpha=0.55$ ), which is in turn in front of the green cluster ( $\alpha=0.4$ ). The use of a warm color in the middle also helps to visually separate the middle cluster from the cool-colored bands on either side of it. In regions where two or three clusters overlap we employ the local model and de-saturate the background color by 50%. We can see that de-saturating the background in these areas preserves the foreground colors and avoids the creation of false hues. Figure 12b shows the same clusters, rendered with the same hues and in the same compositing order, but with opacities assigned in the

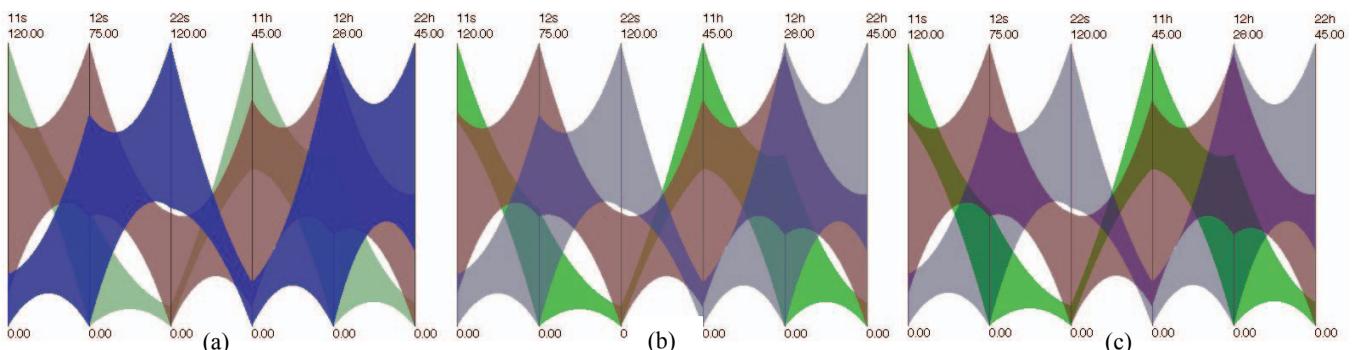


Fig. 12: Illustrative visualizations of a six-dimensional dataset using illustrative parallel coordinates. (a) Ideal visualization with appropriate weightings and color choices, and the use of the local model in overlapping areas. (b) Improper weightings are employed. The blue cluster no longer seems to be in front. (c) The use of improper weightings and the disabling of the local model results in a confusing visualization.

reverse order (i.e., 0.4, 0.55 and 0.7 going front-to-back). We find that it is now virtually impossible to tell that the blue cluster is in front of the others. The visualization further degrades when we no longer employ the local model, which results in the generation of many false hues (Figure 12c). These new shades of purple and greens introduced by mixing are very different from the original cluster colors and could very easily confuse the viewer. They seem to break the clusters into sub-clusters or sub-regions that are not part of the original dataset. Figure 13 compares the effects of cold/warm colors for the perception of front/back ordering. Figure 13a has the blue cluster in front, while in Figure 13b the red cluster is in front. The green cluster is always in the far back. In both cases the opacity was set to 0.5 for all clusters. The local model was employed to avoid false colors. We see that the colorization in Figure 13a gives a more obvious depth ordering that is easier and faster recognized.

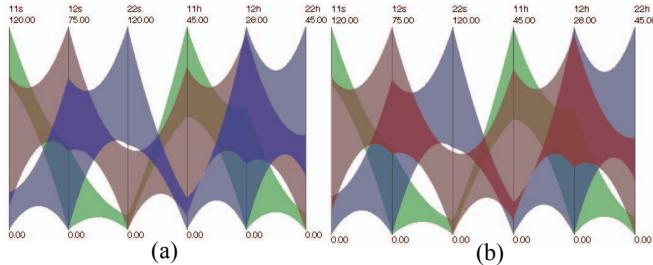


Fig. 13: Effects of cold/warm colors for the perception of front/back ordering. (a) blue layer in front, (b) red layer in front. Opacity  $\alpha=0.5$  for all layers. The green layer is always in the far back, and the local model is used in the overlap regions.

## 8 CONCLUSIONS

We have devised a knowledge-based system that captures a set of prominent guidelines from visual color design as well as insights from human visual perception to encode a set of rules that can be used, in conjunction with data derived from scene analysis and importance, to optimize the assignment of colors in 2D and 3D visualization tasks. Our system is meant to help researchers and practitioners to achieve more task-effective and aesthetic color designs with ease. It seeks to eliminate the trial and error process that comes with picking the ‘right’ colors from a set of millions. We strived to create an interface where users can select the mood of a visualization by picking from a set of suggested hues, with the system then performing the more tedious task of ensuring that it ‘looks good and effective’, assigning the lightness and saturation (vividness) appropriate for the given visualization scene and task. We also provided solutions, and confirmed these via comprehensive user studies, for a variety of existing problems in illustrative and volume visualization, such as the adverse color mixing artifacts when compositing semi-transparent surfaces. In future work we plan to also incorporate information about spatial interactions among scene objects, such as distance, into our scene analysis suite, and to embed perceptual rules on distractor colors [1] to ensure that the importance-based highlighting has the desired effect. We already exploited these types of rules to devise strategies for object highlighting and annotation in color-rich scenes [26]. We would also like to try a numerical optimization algorithm in place of our current randomized algorithm for lightness selection, to see if the colorization results are better or are achieved faster (even though the current strategy is sufficiently interactive).

## ACKNOWLEDGMENTS

This research was partially supported by NSF grant ACI-0093157, NIH grant 5R21EB004099-02, and an equipment grant from the NVIDIA Professor Partnership program. Joachim Giesen and Peter Zolliker were partially supported by the Hasler Stiftung Proj. # 2200.

## REFERENCES

[1] B. Bauer, P. Jolicoeur, W. Cowan. "Distractor heterogeneity versus linear separability in visual search," *Perception*, 25:1281–1294, 1996.

[2] L. Bergman, B. Rogowitz, L. Treinish, "A rule-based tool for assisting colormap selection," *IEEE Visualization*, pp. 118–125, 1995.

[3] B. Berlin, P. Kay, *Basic Color Terms: Their Universality and Evolution*, University of California Press, 1991.

[4] C. Brewer, "Color use guidelines for data representation," *Proc. Section on Statistical Graphics*, pp. 55–60, 1999 (<http://www.colorbrewer.org>)

[5] S. Bruckner, E. Gröller, "Style transfer functions for illustrative volume rendering," *Computer Graphics Forum*, 26(3):715–724, 2007.

[6] *Colorimetry Second Edition*. CIE Publication No. 15.2, 1986 (see also <http://www.brucelindbloom.com>)

[7] T. Callaghan, "Interference and domination in texture segregation: Hue, geometric form," *Perception and Psychophysics*, 46(4):299–311, 1989.

[8] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, Y.-Q. Xu, "Color harmonization," *ACM Trans. on Graphics*, 25(3):624–630, 2006.

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

[10] F. Faul, V. Ekroll, "Psychophysical model of chromatic perceptual transparency based on subtractive color mixture," *J. Opt. Soc. America, A*, 19(6): 1084–1095, 2002.

[11] J. Giesen, K. Mueller, E. Schuberth, L. Wang, P. Zolliker, "Conjoint analysis to measure the perceived quality in volume rendering," *IEEE Trans. Visualization and Computer Graphics*, 13(6): 1664–1671, 2007.

[12] C. Healey, "Choosing effective colors for data visualization," *IEEE Visualization*, pp. 263–270, 1996.

[13] J. Itten. *The Art of Color*. Van Nostrand Reinhold Co., New York, 1961.

[14] H. Kawabata, S. Zeki, "Neural correlates of beauty," *J. Neurophysiology*, 91:1699–1705, 2004.

[15] H. Hauser, L. Mroz, G. Bischli, E. Gröller, "Two-level volume rendering – fusing MIP and DVR," *IEEE Visualization*, pp. 211–218, 2000.

[16] Y. Kim, A. Varshney, "Saliency-guided enhancement for volume visualization," *IEEE Trans. Vis. and Comp. Graph.*, 12(5):925–932, 2006.

[17] H. Levkowitz, G. Herman, "Glhs: A generalized lightness, hue, and saturation color model," *Graph. Mod. Img. Proc.*, 55(4):271–285, 1993.

[18] Y. Matsuda. *Color design (in Japanese)*. Asakura Shoten, 1995.

[19] N. Max, "Optical Models for Direct Volume Rendering," *IEEE Trans. Vis. Comput. Graph.*, 1(2):99–108, 1995.

[20] K. McDonnell, K. Mueller, "Illustrative Parallel Coordinates," *Computer Graphics Forum*, 27(3):1031–1027, 2008.

[21] B. Meier, A. Morgan Spalter, D. Karelitz, "Interactive color palette tools," *IEEE Comp. Graph. & Appl.*, 24(3):64–72, 2004.

[22] R. Motwani, P. Raghavan *Randomized Algorithms* Cambridge Pr, 1995.

[23] P. Moon, D. Spencer, "Geometrical formulation of classical color harmony," *J. Optical Society of America*, 34(1):46–60, 1944.

[24] A. Munsell, *A Grammar of Colors*. Van Nostrand Reinhold Co., 1969.

[25] D. Norman, "Emotion and design: attractive things work better," *Interacion Magazine*, ix(4):36–42, 2002.

[26] N. Neophytou, K. Mueller, "Color-Space CAD: Direct Gamut Editing in 3D," *IEEE Computer Graphics & Applications*, 28(3):88–98, 2008.

[27] B. Rogowitz, A. Kalvin, "Which Blair project: a quick visual method for evaluating perceptual color maps," *IEEE Vis.*, pp. 183–190, 2001.

[28] B. Rogowitz, L. Treinish, "An architecture for rule-based visualization," *IEEE Visualization*, pp. 236–244, 1993.

[29] M. Stone. *A Field Guide to Digital Color*. A.K. Peters, 2003.

[30] N. Svakhine, D. Ebert, D. Stedney, "Illustration motifs for effective medical volume illustration," *IEEE CG&A.*, 25(3):31–39, 2005.

[31] I. Viola, A. Kanitsar, E. Gröller, "Importance driven feature enhancement in volume visualization," *IEEE Trans. on Visualization and Computer Graphics*, 11(4):408–418, 2005.

[32] L. Wang, K. Mueller, "Enhancing volumetric datasets with sub-resolution detail using texture synthesis," *IEEE Vis.*, pp. 75–82, 2004.

[33] L. Wang, K. Mueller, "Harmonic colormaps for volume visualization," (to appear) *IEEE/EG Symp. Volume and Point-Based Graphics*, 2008.

[34] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2<sup>nd</sup> edition, 2004.

[35] M. Wijffelaars, R. Vliegen, J.J. van Wijk, E.-J. van der Linden, "Generating color palettes using intuitive parameters," *Comp. Graph. Forum*, 27(4):743–750, 2008.

[36] W. Wong. *Principles of Color Design*. Wiley, 1996.

[37] R. Zakia. *Perception and Imaging*. Focal Press, 2<sup>nd</sup> edition, 2001.