

Individual Models of Color Differentiation to Improve Interpretability of Information Visualization

David R. Flatla and Carl Gutwin

Department of Computer Science, University of Saskatchewan
110 Science Place, Saskatoon, Canada, S7N 5C9
david.flatla@usask.ca, gutwin@cs.usask.ca

ABSTRACT

Color is commonly used to represent categories and values in many computer applications, but differentiating these colors can be difficult in many situations (e.g., for users with color vision deficiency (CVD), or in bright light). Current solutions to this problem can adapt colors based on standard simulations of CVD, but these models cover only a fraction of the ways in which color perception can vary. To improve the specificity and accuracy of these approaches, we have developed the first ever individualized model of color differentiation (ICD). The model is based on a short calibration performed by a particular user for a particular display, and so automatically covers all aspects of the user's ability to see and differentiate colors in an environment. In this paper we introduce the new model and the manner in which differentiability limits are predicted. We gathered empirical data from 16 users to assess the model's accuracy and robustness. We found that the model is highly effective at capturing individual differentiation abilities, works for users with and without CVD, can be tuned to balance accuracy and color availability, and can serve as the basis for improved color adaptation schemes.

Author Keywords

Color differentiation, assistive technology, visualization, color blindness, modeling, color vision deficiency.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI)

General Terms

Design, Human Factors, Experimentation

INTRODUCTION

The use of color is ubiquitous in modern interfaces and graphical displays. Colors are used to represent a wide variety of meanings including data categories, highlights, continuums, and specific values [24]. Although color is a

valuable tool in representation and visualization, many users in many different situations have difficulty differentiating the colors used on screen. Users with color vision deficiency (CVD, often referred to as color-blindness) are the best known subgroup – and for these users color differentiation can pose extreme challenges – but there are many other reasons for differentiation problems, such as fatigue, glare, lighting conditions, monitor problems, or incorrect display calibration.

Difficulty or inability to differentiate between two colors can have substantial consequences. The problems can range from annoyance and frustration (e.g., if the 'link visited' color in a web browser is indistinguishable from the normal link color), to severe issues of error or safety (e.g., matching colors between a bar chart and its legend, or recognizing an alert color against a background). Although most interface-design guidelines state that redundant encodings should be used in addition to color, there are many examples from information visualization and graphical interface design where this principle is not followed (see Figure 1). Since up to ten percent of the world's population has CVD to some degree [20], addressing the problem of color differentiation could dramatically improve usability for a wide variety of users.

Most existing solutions to the problem involve *re-coloring* – changing some or all of the colors in a visualization to colors that can be differentiated by the user. The main steps in this process are to transform the original display using a model that simulates the user's color perception, then identify regions that are differentiable in the original but not in the transformed version, and re-color these regions so that they are differentiable, using the model to select appropriate colors [23]. The core of this process is the model of the user's color perception. The most frequently used model in re-coloring solutions simulates only certain forms of CVD [4]. If the user's color vision is well described by this model, then existing re-coloring approaches can work well. However, in many cases the standard approach is too broad.

In particular, the standard model does not build a profile that is specific to the user, meaning that several factors are not taken into consideration, such as variations in the user's color perception, non-typical kinds of CVD (of which there are many [1]), or environmental factors such as lighting,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

Copyright 2010 ACM 978-1-60558-929-9/10/04...\$10.00.

monitor settings, or fatigue. By failing to take these issues into account, the current approach does not adequately solve the problem of color differentiation for many users and many environmental situations.

To address this limitation, we have developed a technique called individual color differentiation (ICD) modeling that builds a much more specific model of the user and their environment. The new approach models the user's color-differentiation abilities by testing the user on a series of differentiation tasks, and then uses these results as parameters for the model. There are three main advantages of the new approach:

- It does not require any knowledge about the user's color vision, since parameters for the model come from empirical tests (i.e., calibration);
- It is not limited by a fixed number of predefined profiles, since it builds a model that is individualized to the specific user; this means that it can handle all types of color vision deficiencies;
- It handles environmental effects in addition to factors that are internal to the user's color perception; therefore, it can be used to deal with situations such as glare, lighting, and fatigue, that are not handled by any previous model.

In this paper, we introduce individualized color-differentiation modeling, describe how ICD can be used to improve the process of re-coloring information displays, and report on our evaluation of the approach's efficacy and robustness. ICD is the first model to provide a specialized empirical representation of a user's differentiation abilities, and shows great potential for improving the usability of systems that depend on color to convey meaning.

BACKGROUND AND RELATED WORK

There are four main areas underlying our research: the use of color in information presentation, factors that affect color perception, color-adaptation systems, and existing models of color differentiation.

Color Use in Information Display and Visualization

Color plays a major role in the presentation of visual information, both in everyday graphical interfaces and in specific visualization applications. There are many ways that color is employed, but three main uses that involve color differentiation are encoding categories, encoding continuous variables, and highlighting specific items [21].

Categorical encoding

Visual processing in the human visual system allows rapid identification of colors [24]. Labeling objects with color thus can allow categorical information to be identified quickly and efficiently. In categorical encoding, a unique color is assigned to each category of data, and all representations of this category in the visualization will then employ this color as an identifying characteristic. Color as category is used in a number of information displays (Figure 1), including charts in spreadsheets, 'link taken' encodings in web browsers, syntax coloring in text editors, and tagged messages in email clients.

Healy [6] suggests that a maximum of seven category colors can be used if the luminance of the colors is held constant; with variations in luminance allowed, the number of unique categories is likely to increase. Figure 1a contains an example with twenty categories, which exceeds these maximum category numbers.

Popout, Brushing, and Highlighting

A special case of categorical encoding involves temporary changes to the color of objects that are considered special (Figure 1b). *Color popout* is a visual phenomenon in which color makes elements stand out in an obvious fashion from the rest of the data. The popout color must be sufficiently different from other colors in the visualization for the effect to work: generally a saturated, bright, primary color is used to replace the established element color. Due to the preattentive nature of color, popout allows rapid identification and location of important items.

Brushing is the interactive application of a popout color to a visualization with numerous data points. The user of a visualization marks elements of interest so they remain easily discernible while the data is manipulated (e.g., rotated in a 3D scatterplot).

Highlighting is the use of color to bring attention to an element or region of a visualization. Unlike popout, highlighting does not replace the element color in the visualization, but surrounds the element of interest. As a result, desaturated colors are often used to prevent the highlight from occluding the highlighted item.

Encoding continuous variables

Color is also often used as a means to encode univariate or multivariate data. This process involves the discretization of a continuous data set, where each of the discrete ranges is

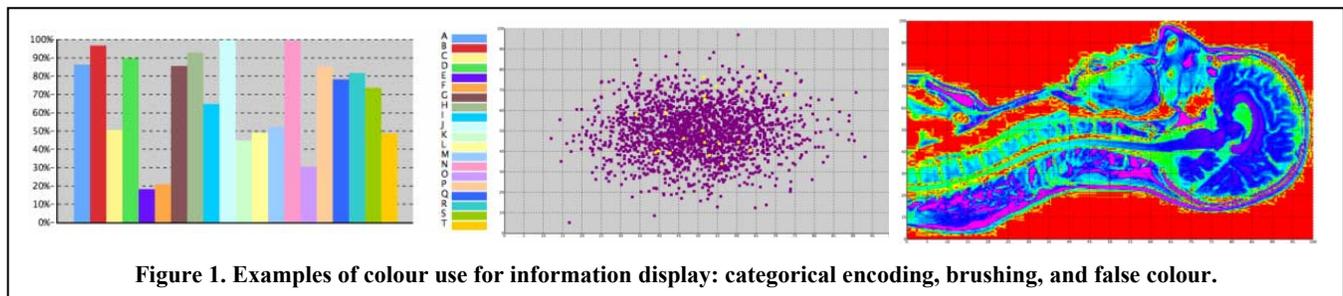


Figure 1. Examples of colour use for information display: categorical encoding, brushing, and false colour.

associated with a given color, and the data set is painted accordingly. For example, the depth of a body of water can be encoded using shades of blue, where darker blues indicate deep water and lighter blues show shallow water. This approach is used in several techniques: false color representations (mapping to hue, as in Figure 1c), continuums (mapping to luminance and saturation), and multidimensional data display (color as dimension).

Mapping an ordinal set of data to a color representation requires the invention of a hue scale. This scale must be learned before the visualization can be used, which can reduce usability [20,24]. Another difficulty with hue scales is simultaneous contrast, which occurs when the perception of a color is influenced by surrounding colors.

Factors that Influence Digital Color Perception

Perception of color is a complex process depending on many factors external and internal to the body. Color is perceived as a distribution of light that enters the eye, and so anything that influences this distribution of light affects color perception. In digital environments, the light source will generally be a monitor; these can vary in their output level of light, as well as the quality of color produced [20].

Once light enters the eye, it is received by the retina and converted to neurological signals, which are sent to the visual centers of the brain for further processing. Color perception can be significantly influenced during this process [1]. For example, old age brings ailments such as yellowing of the lens and cataracts, both of which modify the light entering the eye; in bright lighting, the pupil can also severely restrict the amount of light entering the eye.

Three types of cones generally exist in the retina. Each cone type is sensitive to reception of different parts of the visible spectrum. This absorption is generally centered around a specific frequency for each cone type. CVD results when this frequency is different for certain cones (anomalous trichromacy), or when entire types of cones are missing (dichromacy). In rare circumstances, individuals can be missing two or three types of cones, limiting color perception to shades of grey [5].

Retinopathy occurs when a portion or all of the photoreceptors of the retina die, resulting from diabetes or long-term exposure to styrene. When photoreceptors die, color perception can be drastically altered, reducing blue sensitivity, or resulting in total loss of vision. Some prescription drugs (e.g., Viagra or antidepressants) can also temporarily influence color perception.

Neurological conditions can also influence color vision. As the processing of visual signals flows through a complex procession of cognitive centers, problems along this procession can affect color perception. The presence of depression has been shown to cause the visual field of the afflicted to be shifted to the blue [7]. Brain damage (resulting from stroke or exposure to solvents, for example) can also influence color perception [14].

Color-Adaptation Systems

The idea of adapting colors on a computer display to match the color perception abilities of the user can be linked back to Meyer and Greenberg's work [15], which proposed an early CVD simulation strategy, a computerized color vision test, and the LMS color space. SmartColor [23] is a more recent approach that allows the designer of a visualization to specify color properties for the visualization. These properties then serve as constraints for coloring the visualization in a fashion considerate of CVD.

Many systems have been presented that deal specifically with images (e.g., [8,9,10]). Methods developed for transforming color images to greyscale have been modified to provide accommodation systems for individuals with CVD [16,18], and an accommodation system with an interactive recoloring algorithm has been developed [12,13]. This interactive system was proposed in part to allow the user to explore possible recoloring strategies to find an optimal approach. Some forms of CVD such as anomalous trichromatism are less severe than dichromatism and an interactive system can allow these individuals to guide the recoloring process. Other systems to assist anomalous trichromats have also been developed [19,25].

Although this 'user guided' approach aids the system in selecting the appropriate model to use, it does not achieve the degree of specificity that is part of the ICD model.

Models of Color Perception and Differentiation

Current models of color differentiation are based on an algorithm first presented in 1988 [15] and later updated [22,4]. This algorithm allows the simulation of dichromatic color perception for individuals without CVD. This is achieved with the following steps:

1. $RGB \rightarrow LMS$: Using a pre-defined orthogonal transformation, translate the original image pixel colors (encoded in RGB) into a color representation that encodes color as stimulation levels for the three types of cones (long, medium, short or LMS).
2. $LMS \rightarrow LMS^*$: Manipulate the LMS representation by removing the appropriate wavelength information for the desired type of dichromatic simulation (e.g., long-wavelength for protanopes).
3. $LMS^* \rightarrow RGB^*$: Using the inverse of the orthogonal transformation of Step 1, translate the modified LMS colors back to the original color representation.

To use this simulation algorithm for detecting confused colors, an additional step is required:

4. Compare regions of color in the modified image with the original image. If regions that are different colors in the original image are the same color in the simulated image, then these colors are considered not differentiable (confused).

This approach requires many pieces of information to succeed. The orthogonal transformation requires that the phosphor light emission spectra for the monitor are known.

This varies from monitor to monitor, and particularly between monitor technology such as CRT, LCD, and LED. The monitor also needs to be calibrated both in terms of white balance and gamma. The calibration should result in the pure white of the monitor (255,255,255) being a chromatically pure white, and the intensity of each channel responding in a purely linear manner to input voltage. The orthogonal transformation also assumes a ‘representative’ human color vision system, but variations between humans (polymorphism) is well documented [16]. There are also a myriad of additional factors that influence the color perception of individuals (as described above) such as age, ambient lighting conditions, and the presence of retinal or neurological damage [1,7,14].

The manipulation step requires full knowledge of the type of CVD to be simulated. It does not handle variation in the severity of CVD as in anomalous trichromats, nor does it handle other forms of CVD such as extreme anomalous trichromacy [2], or monochromacy. The manipulation also assumes that the gamut of dichromatic color vision is a proper subset of ‘normal’ trichromatic color vision. To add to this, the algorithm cannot handle the entire RGB gamut, in that the manipulation step occasionally results in some of the RGB colors produced in step 4 being outside of the possible gamut of RGB colors. Lastly, the verification of this system relies on reports of unilateral dichromats, which are of uncertain quality, and the qualitative feedback of only two participants (one protanope and one deuteranope).

This system works well to approximate a simulation of dichromacy for trichromats, but falls short of being a broad, reliable, and complete means of simulating CVD for the purposes of adaptation.

SPECIALIZED MODELS OF COLOR DIFFERENTIATION

Current simulations of color perception are limited in several ways. Our primary goal in this research is a more accurate model of a user’s color perception – in particular, a more accurate model of their ability to differentiate colors on a computer display. Secondary goals are that the model should be easy and cheap to obtain, and should be compatible with existing approaches for color adaptation.

The main drawback of current approaches is that they are not specific enough – either to the user’s particular color abilities or to the environmental factors that affect perception. Our approach has two differences: first, it models each user’s differentiation capabilities individually; second, it uses empirical evidence to build the model.

Empirical models provide a substantial advantage for improving specificity, since they are automatically responsive to all of the factors in the environment that affect color perception, both internal and external to the user. For example, if a user has an atypical type of CVD, and is also using a monitor that is poorly calibrated, traditional simulation models will not fit well; however, an

empirical model built from that user and that monitor will automatically include these factors.

Empirical models gather their evidence through performance or judgment tests that determine what a person can see, rather than a mathematical or procedural simulation of CVD. In the empirical approach, two important questions are how to obtain information to build the model, and what the model will predict.

What information will build the model?

The model’s general technique will be to test the user’s color differentiation capabilities at different parts of the RGB color space, and use these empirical values as the basis for the model. By “the user’s color differentiation capabilities,” we mean the smallest change between two colors, both visible on screen, that the user can reliably identify as different. This amount is called the Just Noticeable Difference (JND) [3].

The most accurate model possible would test the user’s ability with each possible color combination, but sampling must be used to reduce the amount of testing – that is, we will test fewer points in the color space, and interpolate between these points when needed. There are many possible interpolation functions, but based on previous research into perception of sensory stimuli (e.g., [3]), we assume a linear interpolation between samples.

What will the model predict? Differentiation limits

There are two main possibilities for a predictor: first, a decision-based system that takes two colors as input and returns a prediction about whether or not those colors will be differentiable; second, a limit-based system that takes a single color as input, and returns a set of predicted limits that specify the set of colors that can be differentiated from the input color. The second approach is more general, and is the route taken for our model.

There will be six differentiation limits for any color, two for each of the three color channels (R, G, and B). The two limits are the *upper limit* (the value above the input color indicating the limit of what can be differentiated on that channel), and the *lower limit* (the value below the input color marking the limit of differentiability).

For any color in the RGB color cube, these six limits define a box within the cube that contains all colors that cannot be differentiated from the input color. As a simple example, consider the process for a single channel (e.g., Red). For a given red value (e.g., 128), the upper limit for a user determines the minimum red value above 128 that the user can differentiate from 128; the lower limit is similar for values below 128. If the upper limit is 22 and the lower limit is 18, the range of non-differentiable reds (analogous to the box described above) is the range 110-150.

The range (or box) of non-differentiability can then be used to determine whether colors in a display are safe to use: in the above example, if two red values of 128 and 139 were

used in a display, the model could state that these would not be differentiable to the user.

Example of the interpolation model

As stated above, we will use sampling to reduce the calibration required for the model. As a simple example of how interpolation will be used, consider the case of a single channel as introduced above. A model of the user's differentiation ability will be built by testing the user at different points on this channel, and then linearly interpolating between these known points.

The simplest model would be built from two samples, such as the two endpoints: that is, we test the user's upper differentiability limit when $R=0$, and their lower limit when $R=255$, and then use these to interpolate all other limits. Assume that these limits are determined to be 10 and 35. These can be used to determine two pairs of limits, one upper and one lower, as shown in Figure 2. As an upper limit cannot be measured for $R=255$ (ceiling), and a lower limit cannot be measured for $R=0$ (floor), the lines for the limits do not cover the entire scale, but can be extrapolated.

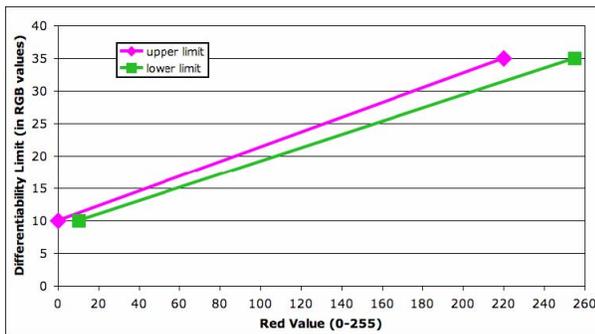


Figure 2. Linear interpolation in a 1-channel 2-sample model.

Dealing with channel dependence

The simple one-channel example described above does not take into account the possible influence of other channels in differentiability. That is, differentiability with $R=0$ may be different when $G=50$ and $B=50$ than it will be when $G=200$ and $B=200$. Therefore, additional samples will be needed for the R channel to account for the influence of different G and B values. We again use linear interpolation to predict for colors in between our input samples.

The example model from above now requires the user's differentiation ability for the R channel with four combinations of G and B : $0,0$; $0,255$; $255,0$; $255,255$. This means that there are eight samples for each channel in a 2-sample model of RGB color.

Full model example

Using these eight points for calibration, we can extend this simple model to a full model. To begin, differentiability limits for each channel must be collected at these eight points giving 24 calibration points, one for each channel at each corner of the RGB cube.

Using these 24 points, linear functions that describe the limits for a channel as we move along an outside edge of the RGB cube can be generated using interpolation. There are twelve of these edges: four describe red channel limits, four describe green channel limits, and four describe blue channel limits.

When the limit box for a color is requested, we process each channel independently and merge the results to get the final box. To process the red channel, the red channel limit functions are used. First, the red value for the color is used to determine the limit in each of the red channel's four functions. These four points define two new limit functions that traverse the green channel dimension. The green value for the color is then used to determine the limit in each of these two new functions. This gives two limit values that are on opposite sides of the RGB cube, spanning the blue dimension, which define another function. Now the blue value for the color is used to determine a limit value along this new function. This value is the red channel limit.

To determine a green limit, we use the green limit functions and the green channel value to establish the four limits (two functions), then the red value to determine two limits (one function), and the blue value to determine the final limit. Blue limits are found by using blue, then red, then green color values to repeatedly interpolate.

It may be noted that the order after the initial selection of four points is irrelevant ($red \rightarrow green \rightarrow blue$ is equivalent to $red \rightarrow blue \rightarrow green$ for red channel limits), and in practice, we calculate each limit using both approaches and cross validate to ensure correctness. Lower limits are determined in the same manner.

EVALUATING THE DIFFERENTIATION MODEL

To explore the effects of different numbers of samples on model quality, and to evaluate the accuracy and effectiveness of the models, we gathered a large set of empirical differentiation data. The study collected extensive data from 16 participants, and investigated three questions:

- Do models with more input samples perform better than those with fewer samples?
- Do the models perform differently for users with CVD compared to those with normal color vision?
- How robust is the model to moderate changes in environmental factors such as light or background?

Methods

Participants, Apparatus, and Differentiation Task

Sixteen volunteers (all male, mean age 26) were recruited from the local community. Eight participants had CVD to some degree (based on the Ishihara plate test [11]), and eight had no indication of CVD. The Ishihara test was performed by the authors in a non-clinical setting, and was used simply to identify the presence or absence of CVD. These tests revealed a mix of protan and deutan effects in the participants with CVD.

The study was carried out in a room with controlled lighting, and used a custom Java application to measure the participant's color differentiation ability. The application presented participants with a series of differentiation tests; the participant's job was to state for each test whether the two colors on the screen were the same or different, by pressing one of two keyboard keys. The participant's responses to these questions were used to empirically determine their differentiability limits.

Each trial presented an 8x6 grid of circles that were randomly colored in one of the two colors for that trial. (Figure 3) A grid was used to reduce retinal after-image effects, by allowing participants to move their gaze around the screen. For each test (i.e., to find out the limits for a particular input color), the system presented repeated trials until the user's limit was determined. The system first presented the furthest possible color from the input color, to determine if the participant could differentiate any colors from the input color; after that, the system used a binary-search strategy to narrow down the color values in each successive trial.

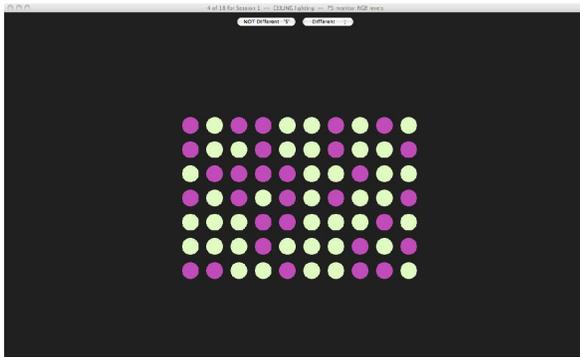


Figure 3. Experimental task. Participants pressed the “;” key if they thought the colors were different, and the “S” key if they thought the colors were the same.

Models with different sample sets

One of the goals of the study was to determine how accuracy is affected by the number of samples in the model. A smaller sample set means a shorter calibration, but also could mean reduced accuracy. In this study, we gathered calibration data from 125 points – that is, from five evenly-spaced points on each of the RGB channels (values of 0, 55, 110, 165, and 220). From this data, we built four models: a 5-sample-per-channel model using all of the data, a 4-sample-per-channel model using the lower four points from the list above, a 3-sample-per-channel model using the endpoints and middle value, and a 2-sample-per-channel model using only the endpoints. (We shorten the names of the models from here on to ‘the n -sample model’)

Robustness

The goal of the empirical modeling approach is to increase the specificity of the model – to be able to automatically include the specific characteristics of the user and their

environment. One potential risk of this approach, however, is an overly-specific model that is too specific to the details of the current environment, and thus not effective when any of those details change. Although there are unlikely to be dramatic changes to the context (e.g., a user's CVD will not switch from one type to another, and a user's monitor will not suddenly switch calibration), there are several ways in which minor changes can occur (e.g., lighting conditions will change over a work day).

We wished to determine how quickly our model's accuracy degrades as the environment changes. Therefore, we tested the model against several additional sets of test colors whose data was gathered under four different environmental conditions. We used the same testing procedure described above. The additional conditions were:

- *Lighting.* In addition to the normal ceiling lighting that was used for the standard tests, we gathered test sets with low lighting (all lights off), and lamp light (a lamp shining on the screen at approximately a 45° angle).
- *Background grey.* In addition to the dark grey background used for the standard tests (grey level 25), we collected test data with two lighter-grey backgrounds (grey levels 128 and 230).
- *Monitor color adjustment.* The display monitor allows the adjustment of color temperature; in addition to the normal value of 75, we also collected data with the monitor's adjustment set to 50 and 100.
- *User fatigue.* We compared results from the start of the testing data with results from the end, to determine whether users' responses change during a session.

Procedure

There were three parts to the study: collection of calibration data, collection of standard test data, and collection of test data under different environmental conditions.

Part 1: Calibration data. The model is calibrated based on empirical samples. In this phase of the study, we gathered the 125 samples described above, from which the different models were built. Tests at each of the 125 color points involved one repetition of the task described above.

Part 2: Empirical data for model testing. To test the accuracy of the various model configurations, we gathered empirical data about the user's actual differentiation limits at eight color points that were distributed through the color cube (no test colors were equal to the calibration colors). Each color was tested 10 times in random order to better estimate the user's true differentiation limits.

Part 3: Empirical data for robustness tests. To test the model's accuracy when environmental conditions change, we also gathered differentiation limits for the eight test colors in each of the situations described above. Each color was tested only once for these data sets.

The study took approximately 2.5 hours to complete, and each participant completed a total of 7600 tests.

To evaluate the model, we carried out two analyses: first, we tested the accuracy of the model using the empirical differentiation data gathered in part 2 of the study; second, we tested the robustness of the model using empirical differentiation data gathered in part 3.

Validating the Model: Accuracy

The first analysis tested the accuracy of the models by comparing the predictions made by the model with the empirical data (which was not used in the formation of the models). As described above, the model predicts differentiation limits for a given color, and we tested the predictions for each of the empirically-determined limits gathered for the eight test colors in Part 2 of the study. We tested four models with different granularities (2, 3, 4, and 5 samples per channel).

For each of the eight colors in our test set, the model predicted the differentiability limits in the R, G, and B channels. Part 2 of the study empirically determined these same differentiability limits, and these are used as the ‘true’ values against which we compare the model’s predictions.

To assess the predictor’s accuracy, we consider the two types of error that the predictor can make – either over or under the true limit value (in addition to the possibility that the predictor is exactly correct). We note that the two types of error are not equal in real-world terms, since an over-estimation will result in false negatives (where the model says that two colors are not differentiable, and the empirical data shows that they are), and an under-estimation results in false positives (where the model predicts that two colors are differentiable but the empirical data shows that they are not). Over-estimation therefore presents much less of a problem in terms of the real-world scenario: it avoids mistakenly allowing non-differentiable colors to remain in the display, but it does reduce the number of colors that can be used.

Based on this analysis, we use the term ‘safe accuracy’ to represent the proportion of predictions that are ‘safe’ – that is, that will not result in a false positive error. This measure (A_{safe}) is the ratio of exact predictions plus over-estimations to the total number of cases.

It is possible to intentionally increase the predicted limits by a constant (called the limit offset), in order to increase the over-estimation and reduce the number of false positives. This moves the distribution of errors towards the ‘safe’ side of the mean. The limit offset can be included as part of the model, and allows us to tune the way that the model trades off false positives and false negatives. We use the limit offset as the measure by which we compare models – in the tests below, we report the offset needed to achieve a safe accuracy of 0.95 (i.e., such that 95% of predictions are either correct or over-estimates).

Figure 4 shows the histogram of prediction errors around the empirically-derived value (4-sample model); errors are normally distributed around the true value, and the model

makes approximately the same number of over- and under-estimates. Table 1 and Figure 5 show our accuracy results. We tested accuracy separately for the two groups of participants (CVD and non-CVD), to determine whether CVD had an effect on the model’s accuracy. We found that higher offsets were needed for the participants with CVD to maintain the same safe accuracy.

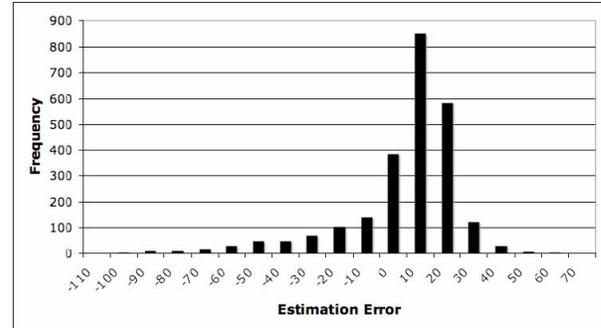


Figure 4. Distribution of errors around the true value for the 4-sample model (predicted value minus true value).

Table 1. Mean accuracy results by model granularity: Off=offset to reach .95 A_{safe} ; S.D.=st.dev. of estimation errors

| | 5-sample | | 4-sample | | 3-sample | | 2-sample | |
|-------|----------|------|----------|------|----------|------|----------|------|
| Group | Off. | S.D. | Off. | S.D. | Off. | S.D. | Off. | S.D. |
| CVD | 52 | 24.3 | 52 | 23.5 | 67 | 27.5 | 81 | 24.8 |
| nCVD | 41 | 18.0 | 43 | 18.7 | 46 | 19.0 | 58 | 15.7 |
| All | 47 | 21.1 | 48 | 21.1 | 57 | 23.2 | 69 | 20.2 |

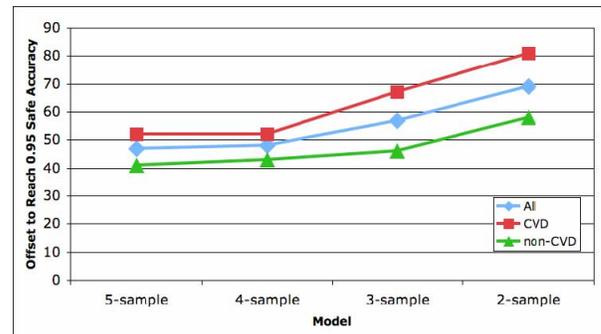


Figure 5. Accuracy (offset needed to reach .95 A_{safe}), by model granularity and participant group.

The explanation for the reduced accuracy is that users with CVD users generally have much larger values for their limits, meaning that the linear interpolation functions hit a ceiling (255) much more quickly than for non-CVD users. The model therefore has fewer values with which to build accurate interpolation functions, resulting in a less accurate model with a higher required offset.

Two further issues to be addressed in our accuracy analysis are the relationship between the limit offset and the safe accuracy, and to what degree over-estimation will reduce the number of colors available to an adaptor system.

Relationship between limit offset and safe accuracy

Figure 6 shows how addition of different limit offsets affect safe accuracy. It is clear that covering the last remaining cases of under-estimation is expensive in terms of the amount of color space that is used; however, it is possible to reach any safe-accuracy level, including 100%.

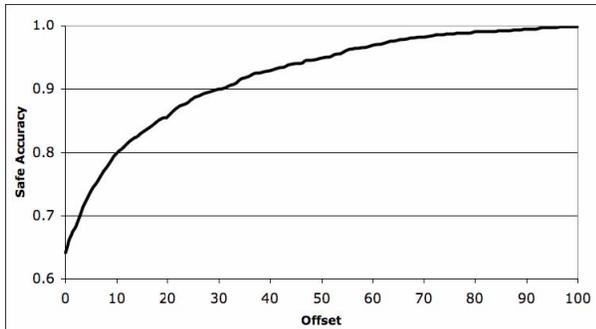


Figure 6. Offset amount vs. safe accuracy (4-sample model)

Reduction in available color space through overestimation

Over-estimation ‘uses up’ more colors than necessary in order to avoid false positive errors, but the number of colors available to an adaptation system depends on several factors. First, in a two-color situation, there is little problem, since even with a limit offset of 100 on each channel, there should still be a large color space remaining within the cube, after removing the input color’s limit box.

If we had a perfect predictor (i.e., no over-estimation required), 0.6% of all possible RGB colors would be removed on average. Using the 4-sample model, with limit offset to maintain 95% safe accuracy, 14.5% of all possible RGB colors are eliminated. Even though the model eliminates many more colors than necessarily required, it still leaves a large set of possible colors to choose from.

The number of available colors even with a very conservative model is large enough to deal with most color tasks. For example, even if only three values per channel can be used, this is still 27 colors, more than enough for the seven maximum that is suggested for categorical encoding [6].

If needed, it is also possible to reduce the degree of over-estimation – this allows more false positives, but preserves a larger color space for a re-coloring algorithm. However, this is unlikely to be a problem except in extreme cases, since even large over-estimations of limits still leaves a very large number of available colors.

An extension to this question is the issue of *co-differentiability* for larger sets of colors, which is discussed in more detail later in the paper.

Robustness of the Model

For the evaluation above, we calculated the limit offset needed to bring the model’s predictions to 95% safe accuracy, and compared this offset value with the offset calculated using the standard test set. By testing that model

against our robustness data (Part 3 of the study), we can determine whether the model’s accuracy degrades when environmental conditions change by moderate amounts. Figure 7 shows our results, using the 4-sample model.

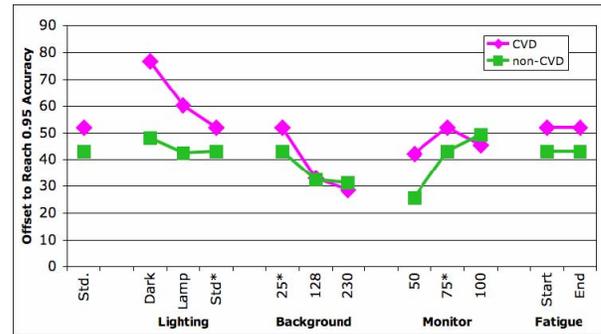


Figure 7. Comparison of offset under different environmental conditions (* = from standard test).

The different environmental factors had different effects on the model’s accuracy, and CVD and non-CVD groups also had different results. There are both increases and decreases to the offset value: increases mean that the model is less accurate in these situations and decreases mean that the model is more accurate. The size of the changes is not dramatic, meaning that the model is not overly sensitive to small changes in the environment. Adding 30 to the limit offset (for non-CVD users) would handle all of the environmental changes that we tested.

Results for users with CVD differ from non-CVD users. There are several potential reasons for this, including the fact that the participants with CVD likely had several types and severities of CVD. In future work we will look at the effects of the different types on these robustness results.

DISCUSSION

Our study shows that individualized differentiation models can effectively represent and predict users’ color differentiation ability. The model has five main strengths:

- automatic sensitivity to specific characteristics of the user and their local environment
- good performance both for individuals with CVD and those with normal color vision
- robustness to moderate changes in the environment (such as light levels or background brightness)
- a tunable offset that allows different balances between false positive rate and color availability
- the availability of different model granularities that provide different prediction accuracy.

In the following sections we discuss several issues in the use and wider deployment of the model.

Calibration in real use

When the model is deployed for real users, it will be calibrated in the same way as described for part one of our study; however, the system will gather data only for the granularity of model (e.g., 3-sample) that the user requires.

The calibration process is lightweight, involving a series of ‘same or different’ decisions. Based on the time needed for each sample in part one of our study, we estimate that calibration will take approximately 4 minutes for a 2-sample model, 14 minutes for a 3-sample model, and 32 minutes for a 4-sample model.

We currently use performance tests at each sample point (i.e., a series of ‘same or different’ tests); however, the time needed for calibration could potentially be shortened by using judgment tests rather than performance tests. In this scheme, users would be asked to move a slider until they felt that there was a just-noticeable difference between two colors on the screen. In future work, we will determine whether judgments are comparable to the performance tests.

It is also possible that the user will need to re-calibrate the model when environmental conditions change beyond what the current limit offset can adequately handle. However, our robustness tests suggest that a single model will be able to handle normal variations in an office environment, at least for the large majority of a user’s color-use tasks.

Using the model with a color-adaptation system

We believe that the individual differentiation model can be used with existing color-adaptation schemes without requiring major changes to those systems. Where an adaptor would previously query a simulation module to determine whether colors are likely to be confused, the system could now use our model either as a decider or as a predictor. As a decider, the model takes two colors and determines if they are differentiable (based on the predicted limits for those colors); as a predictor, the model returns the differentiability box as described earlier. These capabilities should allow our model to be used with a wide variety of other systems, providing them with the benefit of individualized modeling.

In addition, however, the model can also be used as a color-adaptation scheme on its own. Adaptation requires that the model also be able to choose colors that are differentiable; but this requires only a simple extension of the current system to select a color that is outside the limit box of a starting color. This capability is essentially the same as determining a set of n colors that can all be differentiated from one another, which we consider next.

Determining sets of co-differentiable colors

Some tasks (such as matching the colors in a bar chart with colors in the chart legend) require that several colors all be differentiable from one another. The ICD model can be used for this situation, using a process that ‘packs’ the limit boxes for successive colors into the color cube. The general algorithm below specifies the process.

This process can also be used to determine the maximum number of colors that are available for a single task. For example, if a user’s upper differentiability limit is always a

value of 50, then 125 different (and co-differentiable) colors can be used for a particular task.

- ♦ Choose one color as the starting color
- ♦ Add the starting color’s limit box to a do-not-use region
- ♦ For each additional color that must be co-differentiable:
 - ♦ Choose a color just outside the do-not-use region
 - ♦ If the do-not-use region equals the color cube, then fail.
 - ♦ Add that color’s limit box to the do-not-use region

We also note that different tasks can re-use the same color space. For example, if two colors are used to represent ‘link taken’ and ‘link not taken’ in a browser, the same color space could be reused for a bar chart in the same display, since users will be able to separate the colors based on their context.

We also note that *differentiability* of colors is not the only capability that is required for some interactions. For example, *recognition* of colors is a different task that may still be difficult even if colors are differentiable. However, these tasks are all based on differentiability (e.g., if colors are not differentiable, they cannot be correctly recognized).

Using the tunable offset in color adaptation

When the model is used as an adaptation system, the tunable offset becomes particularly valuable, as it can be used to choose colors that will maximize the likelihood of differentiability. For example, if a chart image requires three co-differentiable colors, the model can inform an algorithm that maximizes the distance between these colors based on the user’s perception. In addition, the model can report exactly how accurate its choices are, since the distances between the colors can be used to determine the probability of false positive (exactly as the limit offset is used in Figure 6).

In situations where more colors are required than what can be provided with the specified safe accuracy, the model can report the actual probability of false positive errors based on its attempt to maximize distance between the colors.

Defining colors in terms of differentiability

The capabilities of the empirical model lead to some ideas that could change the way in which color is used in information presentations. One novel possibility extending the SmartColor [23] approach is the idea of letting designers specify a set of requirements for a color, rather than the color itself. These requirements would be specified in a description language (e.g., the way that XML-based languages such as XUUL are used to specify interfaces), and would be related to the function of the color rather than its visual properties. For example, a color could have the requirement that it be differentiable from another color used in the presentation. If a model such as ICD is used, the system could automatically choose colors that will satisfy the requirements, taking into consideration the user and the local environment.

Taking this idea one step further, it is possible to model other perceptual interactions with color and include these in

the specification. For example, color-based ‘popout,’ which works when two colors are sufficiently different from one another, could potentially be defined as a particular amount of separation using the modeling architecture described above. A requirement for popout could then be constructed – and when the system encountered this requirement, it could calculate (based on the user’s individual model) what color difference would provide popout, and choose colors accordingly.

CONCLUSION AND FUTURE WORK

Differentiating colors used in computer systems can be difficult in many situations. Current solutions adapt colors based on standard simulations of CVD, but these models cover only a fraction of the ways in which color perception can vary. We developed an individualized model of color differentiation to improve accuracy of color adaptation. We showed through empirical testing that models can be successfully built for individuals with and without CVD, and with sufficient accuracy for many color-differentiation tasks. The model is tunable to balance prediction accuracy and color availability, requires only a short calibration phase, and is reasonably robust when environmental conditions change. Adaptation systems should be able to use our model immediately, leading to better usability of information visualizations for a wide variety of users.

Our future work will progress in two directions. First, we will refine and extend the model by exploring different interpolation functions, different sampling methods, and different types of input tests; we will also consider other types of color tasks that can be modeled, and will confirm our results with a wider range of participants. Second, we will deploy the model in realistic situations: we will build the color-replacement mechanism described above, and will develop a calibration package to allow wider testing of the approach. Last, we are exploring the range of color tasks in information visualization that could be modeled and specified in a computational description language.

REFERENCES

1. Birch, J. *Diagnosis of Defective Colour Vision*. Butterworth Heinemann, Linacre House, Jordan Hill, Oxford, 2nd ed., (2001).
2. Birch, J. Extreme anomalous trichromatism. In Mollon, Pokorny, and Knoblauch, eds., *Normal and Defective Colour Vision*. Oxford University Press, NY, NY, USA, (2003), 364-369.
3. Booth, D., and Freeman R. Discriminative feature integration by individuals. *Acta Psy.* 84 (1993), 1-16.
4. Brettel, H., Viénot, F., and Mollon, J. Computerized simulation of color appearance for dichromats. *J. Opt. Soc. Am. A.*, 14, 10 (1997), 2647-2655.
5. Cole, B. The handicap of abnormal colour vision. *Clinical and Experimental Optometry*, 87, 4-5 (2004), 258-275.
6. Healey, C. Choosing Effective Colours for Data Visualization. *Proc. VIS '96*, IEEE (1996), 263-270.
7. Heim M., Morgner J. Disturbed color vision in endogenous psychoses. *Psychiatr. Prax.* 28, 6 (2001), 284-286.
8. Iaccarino, G., Efficient Edge-Services for Colorblind Users. *Proc. WWW '06*, ACM (2006), 919-920.
9. Ichikawa, M., et al. Web-Page Color Modification for Barrier-Free Color Vision with Genetic Algorithm. *Proc. GECCO 2003*, (2003), 22-30.
10. Ichikawa, M., et al. Preliminary Study on Color Modification for Still Images to Realize Barrier-Free Color Vision. *Proc. IEEE SMC*, (2004), 36-41.
11. Ishihara, S. *Tests for Colour-Blindness*, 9th ed. Pseudoisochromatic Plates, London, (1950).
12. Jefferson, L., and Harvey, R. Accommodating Color Blind Computer Users. *ASSETS '06*, (2006), 40-47.
13. Jefferson, L., and Harvey, R. An Interface to Support Color Blind Computer Users. *Proc. CHI'07*, (2007), 1535-1538.
14. Lomax, R., Ridgway P., and Meldrum M. Does Occupational Exposure to Organic Solvents Affect Colour Discrimination? *Toxicological Reviews*, 23, 2 (2004), 91-121.
15. Meyer, G., and Greenburg, D. Color-Defective Vision and Computer Graphics Displays. *IEEE CG&A*, 8, 5 (1988), 28-40.
16. Neitz J. and Jacobs, G. Polymorphism of the long-wavelength cone in normal human colour vision. *Nature*, 323 (1986) 623-625.
17. Rasche, K., Geist R., and Westall, J. Detail Preserving Reproduction of Color Images for Monochromats and Dichromats. *IEEE CG&A*, 25, 3 (2005), 22-30.
18. Rasche, K., Geist R., and Westall, J. Re-coloring Images for Gamuts of Lower Dimension. *Computer Graphics Forum*, 24, 3 (2005), 423-432.
19. Ro, Y., and Yang, S. Color Adaptation for Anomalous Trichromats. *I.J. Imag. Sys. Tech.*, 14, 1 (2004), 16-20.
20. Stone, M. *A Field Guide to Digital Color*. Natick, MA. A. K. Peters, Ltd. (2003).
21. Tufte, E., *Envisioning Information*. Graphics Press, Cheshire, Connecticut, USA, 10th ed., (1990).
22. Viénot, F., et al. What do colour-blind people see? *Nature*, 376 (1995), 127-128.
23. Wakita, K., and Shimamura, K. SmartColor: Disambiguation Framework for the Colorblind. *Proc. ASSETS '05*, ACM (2005), 158-165.
24. Ware, C. *Information Visualization - Perception for Design*. Morgan Kaufmann, San Francisco, (2000).
25. Yang, S., Ro, Y., Wong, E., and Lee, J. Quantification and Standardized Description of Color Vision Deficiency Caused by Anomalous Trichromats - Part 1: Simulation and Measurement. *EURASIP Journal on Image and Video Processing*, (2008), 1-12.