Eurographics Conference on Visualization (EuroVis) 2013
B. Preim, P. Rheingans, and H. Theisel
(Guest Editors)

Volume 32 (2013), Number 3

Selecting Semantically-Resonant Colors for Data Visualization

Sharon Lin[⋆], Julie Fortuna[⋆], Chinmay Kulkarni[⋆], Maureen Stone[†], Jeffrey Heer[⋆]

[⋆] Stanford University, USA [†] Tableau Software, USA

Abstract

We introduce an algorithm for automatic selection of semantically-resonant colors to represent data (e.g., using blue for data about "oceans", or pink for "love"). Given a set of categorical values and a target color palette, our algorithm matches each data value with a unique color. Values are mapped to colors by collecting representative images, analyzing image color distributions to determine value-color affinity scores, and choosing an optimal assignment. Our affinity score balances the probability of a color with how well it discriminates among data values. A controlled study shows that expert-chosen semantically-resonant colors improve speed on chart reading tasks compared to a standard palette, and that our algorithm selects colors that lead to similar gains. A second study verifies that our algorithm effectively selects colors across a variety of data categories.

Categories and Subject Descriptors (according to ACM CCS): H.5.m [Information Interfaces]: Misc—Color

1. Introduction

Colors play a central role in data visualization, where they are used to label, measure and enliven data. Appropriate hues help us recognize and discriminate categories. Gradations of luminance or saturation support ordinal and (to a lesser degree) quantitative comparisons.

Colors are also charged with rich associations. Concepts can invoke colors, and vice versa. Common associations in the United States include bananas ↔ yellow, anger ↔ red, and money ↔ green. These associations may be grounded in the physical appearance of objects, common metaphors, or other linguistic or cultural conventions. We use the term *semantically-resonant* to refer to color choices that are evocative of a given concept.

In this paper, we investigate concept-color associations to design effective categorical color assignments for visualization. Theoretically, we note at least two motivating factors for why semantically-resonant mappings may improve chart reading. First, such mappings may aid understanding through semantic facilitation, i.e., they may allow people to use more automated pathways to process value-color associations and require less conscious thought [Baj88]. Second, using resonant colors may improve memory [Ber91]. Practically, improved recognition of category values may reduce the need to consult a legend and may promote future recall.

Can a semantically-resonant assignment of colors to values aid interpretation of data? To address this question, we first conducted a pilot study comparing people's speed on chart reading tasks (comparing values for categories in bar charts) when using semantically-resonant colors or random colors from a default palette. All palettes were designed by the same color expert. Our results showed significant performance benefits for semantically-resonant colors, motivating the present work.

We make two primary contributions: (1) an algorithm for automatic selection of semantically-resonant colors and (2) experimental analyses of chart reading performance under three different color-assignment conditions (expert-crafted, algorithmically-generated, and default order). Figure 1 shows an example of bar charts colored according to these different assignment schemes.

Given as input a set of categorical values and a target color palette, our algorithm maps each data value to a unique color in the palette. We first collect representative images for each value using Google Image Search. Then, we analyze the color distributions of these images to determine affinity scores between data values and candidate colors. Our affinity score balances the *frequency* of a color in the images with *distinctiveness*, or how well the color discriminates between category values. We then compute an optimal assignment of colors to values according to these affinity scores.

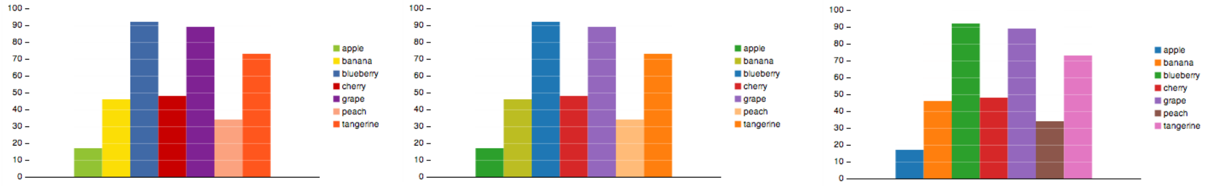


Figure 1: *Bar charts depicting fictional fruit sales. The charts use color assignments from an expert (left), our algorithm (center), and a standard palette (right). The first two charts use semantically-resonant colors to represent data values.*

Our algorithm selects among colors present in a candidate palette, rather than generating a color palette from scratch. The candidate palette can be chosen to enforce desirable perceptual properties, including a proper luminance range, equivalently salient colors, color distinctiveness and nameability [HB03, Sto08, HS12]. We assume an appropriate palette has been provided; the algorithm focuses on determining a semantically-resonant assignment.

We then present results from two experiments that evaluate our algorithm and assess the benefits of semantically-resonant colors. The first experiment measures participants' speed on a set of chart reading tasks with a small number of categories. Participants see expert, algorithmic, or ordered color assignments. Both the algorithmic and ordered assignments use the 20-color categorical palette designed for Tableau, a commercial visualization tool. Expert color assignments were hand-crafted by the creator of the Tableau palette and not constrained to a fixed candidate color set. This allows us to estimate an upper bound on the benefits of semantically-resonant colors. We find that both expert and algorithmic assignments significantly improve performance over ordered assignments, and that algorithmic assignments have comparable performance to expert quality overall. The second experiment verifies that the benefits of algorithmic assignment generalize to a wider set of categories.

2. Related Work

2.1. Color Names and Cognition

Psychologists have investigated how color names contribute to semantic interference or facilitation. For example, linguistic patterns in color naming may affect which color shades are confused in memory [RDD00, WB07]. Contrasting textual and color cues produce strong interference (the Stroop effect) [Mac91]. When color names are printed in a conflicting color (e.g., red or yellow) subjects take longer to name them. Incongruent color-related words (e.g., fire and grass) also cause interference [DA72]. Interference is not observed when colors are semantically-resonant with the text.

2.2. Modeling Color Associations

In the closest related work, Havasi et al. [HSH10] investigate how to select colors for words and phrases that describe a concept. They introduce an algorithm that generates colors for a particular word by interpolating between known word-color associations. Their algorithm maps a single value (word or phrase) to a single best color, and does not consider mappings to multiple valid colors (e.g., apples may be red or green). In this paper, we select among multiple possible color assignments for a value and examine how discriminative colors are across a set of values.

A related topic is modeling associations between colors and names. Some researchers fit statistical models to human judgements of color-name associations [BTBV02, Mor03, CSH08, MMW10, HS12]. Others learn these associations by analyzing images retrieved from search engines [VDWSVL09, SS12], using topic models such as Probabilistic Latent Semantic Analysis (PLSA) [VDWSVL09] or supervised Latent Dirichlet Allocation [SS12]. Additional heuristics (e.g., saliency detection and outlier removal) increase model robustness [SS12]. The goal is to predict a color name label given an image pixel or region. Although our method also queries image search engines, our goal is to assign semantically-resonant colors to categorical data.

2.3. Color Palette Design Tools

Previous work on color design for data visualization falls into practical guidelines and interactive systems for informing color choices. These guidelines are based on data type, number of classes, and perceptual constraints. For example, large differences in luminance and saturation may suggest an ordering in colors and should be avoided for qualitative palettes [HB03]. Colors should be well separated [Hea96] and should not compete with each other [Tuf90]. To safeguard against these pitfalls, by default our method selects colors from the Tableau 20 palette, which was designed specifically for visualization applications.

Interactive systems and algorithms to guide color choices also exist, but they do not consider semantic associations between data values and colors. Prior work has optimized color mappings based on spatial frequency [BRT95], perceptual visibility [LSS12], color harmony [WGM*08], and display energy consumption [CWM09]. Rheingans and Tebbs [RT90] introduced a tool allowing users to manipulate color mappings to visually explore and filter data. Lastly, other work has focused on generating palettes for artistic rather than visualization applications [MSK04, OAH11].

2.4. Crowdsourcing Graphical Perception

Online crowdsourced experiments are attractive for their increased scalability and reduced cost, and have been used successfully in a number of perceptual experiments [HB10, KZ10, TGH12] and color naming studies [Mor03, CSH08, MMW10, HS12]. Heer and Bostock [HB10] demonstrate the validity of crowdsourcing graphical perception experiments by replicating prior laboratory experiments on Amazon Mechanical Turk. Their crowdsourced results are consistent with laboratory findings, albeit with higher variance. This higher variance may be due to a diversity in factors such as display size, viewing distance and lighting, which may more accurately reflect real users' environments.

3. Selecting Semantically-Resonant Colors

Our algorithm for semantically-resonant color assignment takes as input a set of categorical values and a set of candidate colors. It then outputs an assignment from categorical values to unique colors in the candidate set. We focus on mapping to colors in a preselected palette to ensure desirable perceptual properties. In this paper, we use the Tableau 20 palette (Figure 2), a set of 20 colors designed for visualizing categorical data.

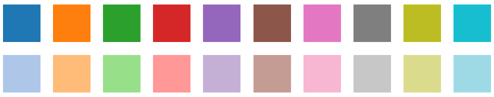


Figure 2: *The Tableau 20 color palette.*

For our studies we asked a color expert (the creator of the Tableau 20 palette) to craft semantically-resonant color assignments. The expert started by finding relevant images for each category set to identify representative hues. Different kinds of images worked better for different categories. For example, for fruits and vegetables, illustrations were more useful than photographs. The expert then refined colors by hand to optimize saliency and distinctiveness. When categories had multiple valid colors, she chose colors based on the color associations of other values in the set so that the color assignment had a more discriminating range of colors.

Our algorithm follows a similar process, and works in two stages. First, in the data collection stage, it retrieves relevant images and computes aggregate color histograms for each categorical value. Second, in the color assignment stage, it uses the color histograms to calculate color-value affinities that balance the probability of a value taking a given color with how well a color discriminates between values. The algorithm then assigns values to colors to maximize the affinities. We tuned the algorithm parameters using color assignment judgments from workers on Mechanical Turk.

Separating data collection and assignment has two advantages. First, data is collected for each categorical value independently of others, allowing data to be reused. Second, in visualization tools, users may switch between different styles of palettes, such as bold or pastel. Having a separate assignment phase enables users to quickly swap palettes, without having to collect data or compute aggregates again.

3.1. Data Collection

In the data collection stage, the algorithm retrieves images for each value and computes aggregate histograms.

Searching for images. We use Google Image Search to find images for each value. This approach has proven successful for learning color name associations [VDWSVL09, SS12]. We use two image queries: one with the original value text and one with "clipart" appended. Multiple queries diversify the types of images. As our color expert noted, illustrations are more useful than general images for some categorical values. For example, in the United States people commonly associate money ↔ green, which is represented in clip art. However, in photographs currency bills are closer to gray. The regular query is useful for values that are not depicted in clip art, or for which the clip art query returns irrelevant images. We explored other query expansions, such as appending "color", but we found these tend to return more irrelevant images (e.g., of coloring books or rainbow-colored images). The Google Image Search API limits the number of image results to 32 per query. Filtering broken links and grayscale images results in 29.5 images per regular query and 27.3 images per clip art query on average.

Computing aggregate color histograms. Our algorithm computes aggregate color histograms for each image query type. For each categorical value, we create two color histograms, one for the regular query and one for "clipart". We tabulate pixel colors across all images retrieved by a query using $5\times5\times5$ unit bins in CIELAB space. This bin size corresponds to a radius of approximately one just noticeable difference (JND) [Sha02].

Retrieved images often contain white or black solid backgrounds that are unrelated to the data value. Thus, we remove these backgrounds from images having at least two connected components. Adjacent pixels are part of the same connected component if their colors are within a distance threshold τ. We consider an image to have a white (or black) background if 75% of its border are within τ of white (or black). We set τ to 3 CIELAB units (slightly larger than 1 JND). We found that advanced background subtraction algorithms such as GrabCut [RKB04] and object saliency detection [SS12] do not usually improve pixel selection, and sometimes omit useful information. For example, such algorithms may ignore "forest" image backgrounds with many leaves and signature background colors from business logos.

3.2. Color Assignment

In the color assignment stage, the algorithm computes color-value affinities that balance the probability of a value taking on a color with how well that color discriminates between values. It then optimally assigns values to unique colors according to these affinities. We tune all parameters in this stage using cross-validation on a training set of crowd-sourced semantically-resonant colors (Section 3.3).

Color probabilities. We calculate the conditional probability of a candidate color c given a categorical value v and corresponding histogram T by applying kernel density estimation (KDE) to the histogram (Equation 1). KDE is a non-parametric method of estimating probability densities that is robust to histogram bin size variation and can handle categorical values with multimodal color distributions. We ignore color bins close to white, as very light and desaturated colors are common across images but less useful for visualizations on a white background.

$$p(c|v,T) \propto \sum_{\substack{b \in T \\ ||b - white||_2 \geq w_t}} T(b)\, \exp\left(-\frac{1}{2}\left(\frac{dist(b,c)}{\sigma}\right)^2\right) \tag{1}$$

In Equation 1, w_t is the distance threshold from white, and $dist(b,c)$ is a color distance metric. Different distance metrics might be used here. We use color name cosine distance [HS12], as it respects categorical color boundaries. Based on cross-validation, we set $w_t = 20$ units in CIELAB space and set $\sigma = 0.2$ (see Section 3.3 for details).

This process results in two color distributions, one for each type of histogram (regular T_r and "clipart" T_c), from which to infer color information. Recall that some values benefit from the "clipart" query images but others do not. We assume that the histogram with stronger color associations for a particular value is more relevant to that value, and thus contributes more to its overall color distribution.

To measure relevance, we look at entropy, a common information-theoretic measure of the randomness of a distribution. The entropy of colors for a histogram is:

$$H(C|v,T) = -\sum_{c \in C} p(c|v,T) \ln p(c|v,T) \tag{2}$$

where C is the candidate set of colors, v is the given categorical value, and T is the color histogram. If $p(c|v,T)$ is 0, we skip contributions from c in the summation. The larger the entropy of colors for a given histogram, the more random the distribution and the weaker the color association. Therefore, we weight the two probability distributions by the inverse entropy, $H(C|v,T)^{-1}$.

The final probability of a color c given a value v is then:

$$p(c|v) \propto \max(sat(c),t) \cdot \left[\frac{w_c\, p(c|v,T_c)}{H(C|v,T_c)} + \frac{(1-w_c)\, p(c|v,T_r)}{H(C|v,T_r)}\right] \tag{3}$$

where $t = 0.1$ is a minimum saturation threshold and $w_c = 0.7$ is the prior bias towards the clip art histogram. The terms $\frac{w_c}{H(C|v,T_c)}$ and $\frac{(1-w_c)}{H(C|v,T_r)}$ are the relative contributions each histogram type makes towards the final color distribution for that value. Thus, the algorithm favors the image query source that has more consistent color associations, which may change depending on the value.

Our formulation weights the probability based on the saturation of the candidate color. We prefer candidate colors if they are closer to histogram bins with high counts and have higher saturation. High-saturation colors tend to be more discriminative, and mixing different levels of saturation can imply order misleadingly [HB03].

Color-value affinities. We compute color-value affinities using color probabilities. Our affinity score promotes colors that are common in images but also distinguish values from others in the set. The affinity of a value v to a color c is:

$$\text{affinity}(c,v) = \frac{p(c|v)}{H(V|c)} \tag{4}$$

where V is the set of categorical values and $H(V|c)$ is the entropy of values given the color c, defined similar to Equation 2. Intuitively, $H(V|c)^{-1}$ indicates how well a color discriminates values from each other.

Several other choices exist for defining an affinity score. We experimented with $p(c|v)$ by itself as well as point-wise mutual information between colors and values. In our tuning and cross-validation tests, $p(c|v)$ and $\frac{p(c|v)}{H(V|c)}$ performed comparably but we find the balanced metric provides qualitatively better color separation.

Lastly, we compute a color-value assignment that maximizes the sum of affinities using the Hungarian method [Kuh55]. Intuitively, this method resolves color-affinity conflicts by compromising based on the strength of a value's affinity to each color. If two values v_1 and v_2 have a strong affinity for color c, but v_1 has no alternative colors while v_2 also has moderate affinity for an alternative color, then v_1 will be assigned to c and v_2 to the alternative.

3.3. Algorithm Tuning and Validation

Our algorithm introduces several parameters that need to be set to achieve good color assignments. We tune these parameters based on a collection of categorical value sets and associated semantically-resonant colors. In this tuning process we first gather sets of categorical values. We then collect color assignments for the data from Mechanical Turk. The collected data is split into training and test sets. Finally, we choose parameters based on how well the resulting automatic assignments overlap with Turkers' choices in the training set. We evaluate the performance of the tuned model by computing the overlap of automatic assignments with data in the test set. The overlap between two color assignments for the same set of categorical values is defined as

$$\text{overlap}(A,B) = \sum_{v \in V} [A(v) == B(v)] \tag{5}$$

where A and B are color assignments and V is a set of values.

3.3.1. Gathering Categorical Value Sets

We collect 40 categorical value sets for training and testing. The sets were gathered from online graphs used by Savva et al [SKC*11]. The graphs come from various sources on the web and so reflect a reasonable sampling of different types of categories that people visualize or find interesting.

Across domains, categorical values can vary significantly in their strength of color association. For example, categories about crops (wheat, corn, *etc.*) are more strongly associated with colors than are continents. In addition, some category sets contain values that map well to unique palette colors, while others have values that compete for the same color. For example, different oil types (canola, sunflower, and olive oil) all map to yellow. These characteristics influence how well a given categorical value set maps to semantically-resonant colors, or in other words, how *colorable* the value set is.

We anticipate that both the algorithm and Turkers will choose better semantically-resonant colors for more colorable value sets. Thus, when gathering the 40 value sets, we (subjectively) aimed for a range of colorability with 28 reasonably colorable sets and 12 non-colorable sets. A few sets were manually shortened by removing non-colorable members to make them more colorable and understandable. For example, a 14-value set of gambling activities was shortened to 7 by removing values such as “Pulltabs” and keeping better-known activities such as “Horse racing”.

In each task, we asked Turkers to create color assignments using the Tableau 20 palette for 10 value sets (7 colorable, 3 non-colorable). The instructions encouraged them to pick the most “representative” color for each value, where no two values can take on the same color. We also asked Turkers to rate the strength of color-value association on a Likert scale from 1 (Not associated at all) to 5 (Very strongly associated). We treat the Likert ratings as an indicator of colorability.

The order of the value sets as well as the values within a set were randomized. Tasks (Turk HITs) were limited to the United States and priced at $1 USD. We gathered 25 palettes for each value set. Figure 3 shows example sets (1 colorable, 1 non-colorable) with their crowd-chosen colors.

3.3.2. Tuning Based on Crowdsourced Palettes

Given a dataset of crowdsourced color assignments, we select algorithm parameters that best match the crowdsourced data. From our 40 value sets we set aside 10 sets evenly spread across the colorability ratings as test data. The other 30 sets are used for training. We tune parameters based on 10-fold cross validation on the training data, using average overlap (Equation 5) with Turkers as the evaluation criteria.

During tuning we also experimented with different color distances in Equation 1. We considered the current standard for perceptual color distance, CIEDE2000 [SWD05], and color name cosine distance [HS12]. Empirically, we find that color name cosine distance gives better cross-validation results on the training set according to our overlap criteria. Qualitatively, color name cosine distance respects color boundaries better than CIEDE2000, which might replace black with blue or yellow with green due to proximity.

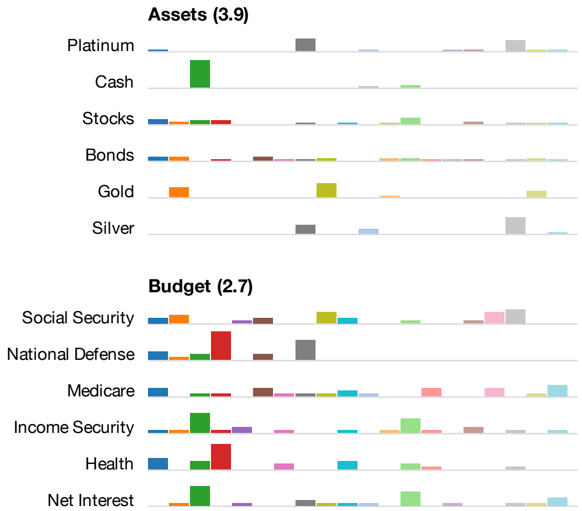


Figure 3: *Example categorical value sets and the counts of colors chosen by Turkers. Numbers in brackets are the mean colorability ratings. The first is an example of a more colorable set (mean rating: 3.9) while the second is less colorable (mean rating: 2.7)*

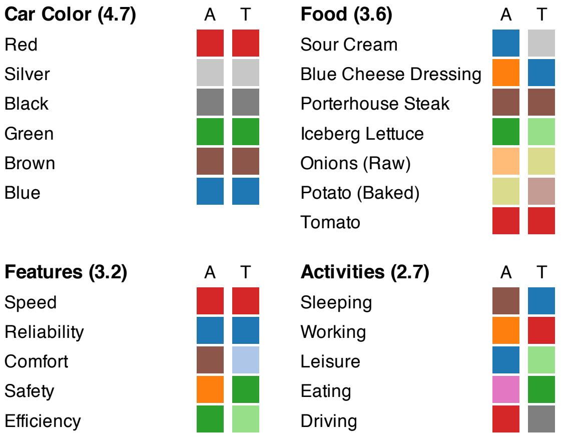


Figure 4: *Category sets ordered by mean colorability rating. Ratings are shown in parentheses. Column A contains colors selected by our algorithm; Column T contains the Turker-chosen colors with highest overlap.*

3.3.3. Average Assignment Overlap with Turkers

After tuning on the training set, we run the algorithm on the test set of categorical value sets. Figure 4 shows a few

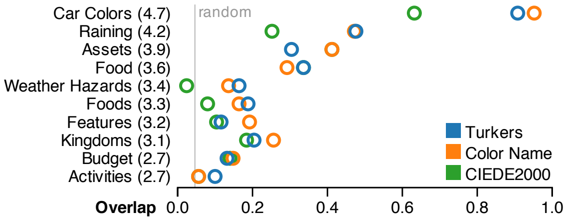


Figure 5: *Between-subject (Turkers) and algorithm-subject agreement using color name distance and CIEDE2000 distance, as measured by average overlap. The gray line shows the expected performance of randomly-selected colors. The mean colorability ratings are in parentheses.*

of the algorithmically generated assignments alongside the most representative Turk assignment according to overlap.

Figure 5 compares the algorithm's overlap with Turk assignments. To compute Turker agreement (blue), we calculate the average overlap of one Turk assignment to all other Turk assignments for a given categorical value set. As the colorability rating of the categorical value set increases, the average performance of all assignments also tends to increase. Color name distance (orange) results in improved performance on the test set compared to CIEDE2000 distance (green). On average, the performance of the algorithm appears comparable to Turkers.

4. Experiment 1: Semantically-Resonant Colors

Our first experiment had two goals: (1) to verify that semantically-resonant colors lead to improved graph reading performance, and (2) to compare the performance of our algorithm to both expert-chosen color assignments and sequential assignment from the Tableau 20 palette. We had two primary hypotheses:

H1: *Semantic resonance improves performance*. Both expert-chosen and algorithmically-chosen semantically resonant colors will improve graph reading performance.

H2: *Expert-chosen assignments outperform others*. An expert can draw on a wider amount of knowledge and produce better semantically-resonant colors than our algorithm.

We tested these hypotheses through an experiment conducted on Amazon Mechanical Turk. We asked participants to answer questions using category-colored bar charts. Questions were representative of common graph-reading tasks (e.g., "which category has a larger value?"). We colored the bar charts using either the expert-chosen, algorithmically-chosen, or sequential colors.

Participants. 144 US-resident workers (77 female) on Mechanical Turk participated in the experiment. Four participants reported they were color-blind, and their data was discarded from analysis. Participants were paid $2 USD.

Method. The experiment followed a mixed between- and within-subjects design, summarized in Table 1. We used a mixed design (rather than a pure within-subjects design) to reduce participant fatigue and ensure consistency with our second experiment, which involved only two assignment types. For a single block of trials we showed participants a colored bar chart and asked 30 binary forced-choice questions (see Figure 7). Following the methods of Cleveland & McGill [CM*84], chart values were uniformly distributed between 5% and 95% on a scale of $0-100\%$. All individual and aggregate values referenced by the experimental prompts were separated by at least 5%.

Each trial involved one of three *question types* (within-subjects variable). Participants compared individual bars (A vs. B) or combinations of bars (A vs. $B+C$, $A+B$ vs. $C+D$) and were asked to report which of the quantities is larger. We used these three question types to measure the effect of color when different amounts of cognitive processing is required. These question types follow prior experiments by Spence & Lewandowsky [SL06], who found them effective for proportional comparisons.

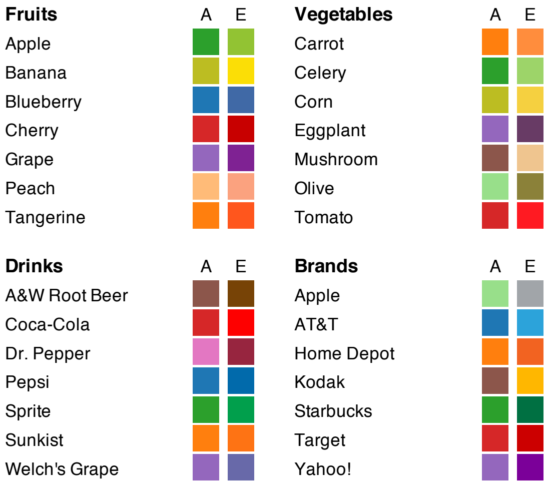
Table 1: *Independent variables in Experiment 1.*

Variable	Levels
Question Type	"Which is larger, A or B?" "Which is larger, A or B+C?" "Which is larger, A+B or C+D?"
Category Type	Concrete or Iconic
Assignment Type	Ordered, Expert or Algorithm

At the start of the experiment, participants completed training with 10 sample questions about data unrelated to later trials. Participants then performed two blocks of trials. Each subject saw questions from one *category type* (between-subjects variable), and from two *assignment types* (within-subjects). For the category type, subjects either saw *concrete* values (fruits and vegetables) or *iconic* values (company brands and soft drinks). We used two category types to reflect the fact that some categories have non-arbitrary color associations (cherries are physically red), while others have arbitrary associations (Coca-Cola's branding is red, but the drink is not). For the assignment type, participants saw either *Expert/Ordered*, *Expert/Algorithm*, or *Algorithm/Ordered* combinations.

The expert color assignments were crafted by the designer of the Tableau 20 palette. We use Tableau 20 as both the candidate color set for our algorithm and for the ordered (sequential assignment) condition. We did not constrain the expert to choose colors from Tableau 20, so as to determine an approximate upper-bound for the benefits of semantically-resonant colors. Categories and corresponding expert and algorithm-chosen colors are shown in Figure 6.

In the first block of trials, each participant was shown 30 questions from the first category (10 of each question type).



Fruits	A	E	Vegetables	A	E
Apple	🟩	🟩	Carrot	🟧	🟧
Banana	🟨	🟨	Celery	🟩	🟩
Blueberry	🟦	🟦	Corn	🟨	🟨
Cherry	🟥	🟥	Eggplant	🟪	🟪
Grape	🟪	🟪	Mushroom	🟫	🟫
Peach	🟧	🟧	Olive	🟩	🟩
Tangerine	🟧	🟧	Tomato	🟥	🟥

Drinks	A	E	Brands	A	E
A&W Root Beer	🟫	🟫	Apple	🟩	⬜
Coca-Cola	🟥	🟥	AT&T	🟦	🟦
Dr. Pepper	🟪	🟥	Home Depot	🟧	🟧
Pepsi	🟦	🟦	Kodak	🟫	🟨
Sprite	🟩	🟩	Starbucks	🟩	🟩
Sunkist	🟧	🟧	Target	🟥	🟥
Welch's Grape	🟪	🟪	Yahoo!	🟪	🟪

Figure 6: *Color assignments for categorical values in Experiment 1. (A = Algorithm, E = Expert)*

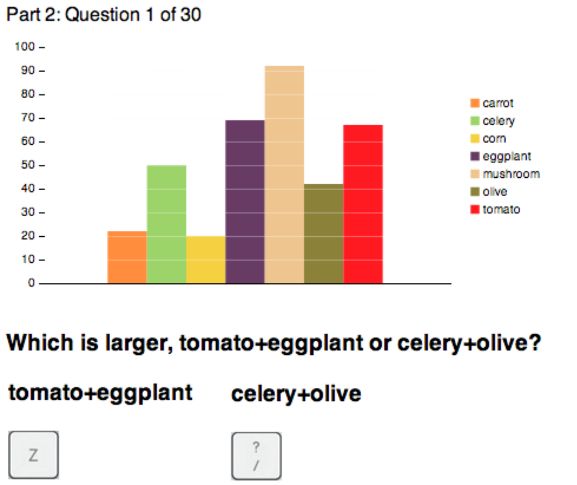


Figure 7: *Experiment trial for an A+B vs. C+D question.*

This was followed by the second block of trials: 30 questions from the second category (10 of each question type). For instance, a participant might see 30 questions on company brands, followed by 30 on drinks. Questions were shown sequentially, with the next question appearing immediately after the participant answered the current one, both to mitigate participants task-switching out of the experiment, and to get better timing information. Both assignment type order and question order were counter-balanced.

We measured both accuracy and response time as dependent variables. Response time was the primary measure of interest; accuracy was recorded to ensure task compliance. All timing information was captured on the client using JavaScript to avoid inaccuracies due to round-trip server delays. Lastly, participants could only respond using the keyboard to eliminate mouse movement delays.

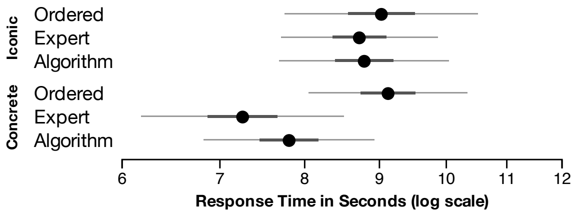


Figure 8: *Experiment 1 results. Points depict means of log response times, by condition. Error bars show 50% (thick) and 95% (thin) confidence intervals.*

4.1. Results

As is common with response times, we found the distribution to be log-normal. We subsequently log-transform task completion times for use with ANOVA and mixed-effects models. An initial, repeated-measures ANOVA shows a significant effect of assignment type on question answering time ($F(2,137) = 46.27, p < 0.001$).

We then analyzed the data using a linear mixed-effects model, using each participant as a random effect with a fixed intercept. The assignment type was contrast-coded as *resonant* or *ordered*, and as *expert* or *algorithmic*. These contrasts let us test whether resonant assignments help (H1), and if expert assignments outperform algorithmic ones (H2). We report the results from the mixed-effects model built with the `lme4` package in R. We report the conventional degrees of freedom (based on the experimental design), model t-values, and p-values from a Markov chain Monte Carlo simulation.

4.1.1. Semantically-resonant colors improve speed

We found a main effect for semantically-resonant assignments. Both expert and algorithm assignments improve performance across category types ($t(126) = -2.24, p < 0.05$), supporting H1. The model also has a significant interaction effect of resonance and category type ($t(126) = -4.88, p < 0.05$); resonant assignments helped most for the concrete, non-arbitrary categories.

4.1.2. Expert assignment outperforms algorithm for concrete categories

The type of resonant assignment had no significant main effect ($t(126) = 0.64, p > 0.3$). Overall, expert-chosen colors did not improve performance more than algorithmic assignment. However, there is a significant interaction effect of resonant assignment and category type ($t(126) = -3.89, p < 0.05$). Expert-chosen colors appear to outperform the algorithm for concrete categories (Figure 8).

4.1.3. Question type moderates resonance gains

We found one significant interaction between question type and assignment type. Resonant color assignments of both kinds helped less with $A{+}B$ *or* $C{+}D$ questions ($t(126) =$

$2.03, p < 0.05$). This may be because a large fraction of the time for these complex questions is spent in computing the right answer, rather than on identifying different data values.

4.2. Discussion

Speed. Across categories, semantically-resonant assignment had a small but significant effect on response times. The effect sizes for log(time) using Cohen's d measure are $d = 0.15$ for the algorithm and $d = 0.17$ for the expert colors, compared to ordered assignment. On average participants took 9.1 seconds per task using ordered assignments and 8.2 seconds and 8.1 seconds for algorithmic and expert assignments, for an average improvement of 1 second. Much of these savings come from the concrete categories; iconic categories have little or no effect (approximately 0.25 seconds). This difference is not significant via a Tukey HSD test.

Though small, these effects can have important practical benefits. Time savings are measured in terms of a single comparison task. Visualization viewers often engage in many such comparisons both within and across plots. Small gains in low-level recurring tasks may both speed chart reading and reduce overall cognitive load.

Iconic vs. Concrete categories. Our current experiment shows a surprising difference in performance across iconic vs. concrete categories, but does not offer an explanation. These differences may be particular to the assignments and categories used, as we were constrained by the number of expert-designed color assignments available. We look at a larger set of diverse categories in Experiment 2.

5. Experiment 2: Wider Assessment across Categories

The goal of our second experiment was to verify that the performance benefits of algorithmic assignment generalize to a wider set of categories. We used the test set of categories from Section 3.3.2, sampled from the ReVision visualization corpus [SKC*11] and spread across colorability ratings (the list of categories is included in the supplementary material). The number of values per category ranged from 4 to 8, with a mean of 5.7. Experiment 2 follows a design similar to Experiment 1; we describe any salient differences below.

Participants. 302 US-resident workers (224 female) participated on Mechanical Turk, and were paid $1 USD. No participants reported being color-blind.

Method. We used a between-subjects experiment design. Each participant saw graphs for one category colored using an ordered or algorithmic assignment. Subjects saw up to 10 questions for each question type; smaller sets had fewer questions of the form $A+B$ vs. $C+D$. Participants saw 25 questions, for an average of 8.3 questions of each type. Participants saw at least one question from each type. We used the same interface, candidate color palette (Tableau 20), and question ordering scheme as Experiment 1.

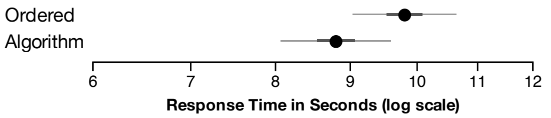


Figure 9: *Experiment 2 results. Points depict means of log response times. Error bars show 50% (thick) and 95% (thin) confidence intervals.*

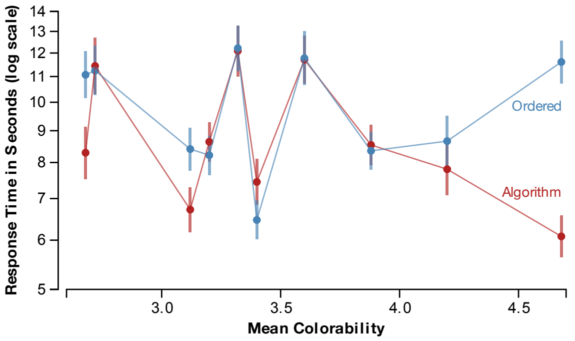


Figure 10: *Experiment 2 results. Points depict means of log response times for each category, organized by colorability. Error bars show 50% confidence intervals.*

5.1. Results

A repeated measures ANOVA shows significant differences in (log transformed) task completion times between ordered and algorithmic assignment ($F(1,300) = 5.95, p < 0.05$). Post-hoc analysis below uses a mixed-effects model with participant as a fixed random-effect and assignment type, question type and interactions as covariates. We report results similarly as in Experiment 1.

5.1.1. Algorithmic assignment improves speed

With a model that includes the interaction between Assignment and Question type, we found algorithmic assignment has a main effect on task completion time ($t(296) = 2.16, p < 0.05$, Cohen's $d = 0.16$; the degrees of freedom are reduced because we added an interaction term). Algorithmic assignment leads to faster performance, as shown in Figure 9. In this experiment, we found no interaction between the assignment and the question type ($t(296) = 1.17$ for A *vs.* $B+C$ and $t(296) = 1.52$ for $A+B$ *vs.* $C+D$).

5.1.2. Algorithmic assignment preferentially benefits highly colorable categories

A mixed-effect model that includes both the above interaction terms and colorability (Section 3.3) as a covariate has a significant interaction effect of mean colorability rating and color assignment (Figure 10): algorithmic assignment reduces time for more colorable categories ($t(294) =$

$-2.20, p < 0.05$), while ordered assignments have no such benefit ($t(294) = -0.29, p > 0.3$).

Are the performance gains associated with algorithmic mapping limited to highly colorable categories? If we omit the two most colorable category sets from our data ("cars" and "raining"), we find that algorithm assignments do reduce completion time, but the effect is no longer significant ($t(204) = -1.54, p = 0.12$; the number of observations is reduced, which also reduces conventional degrees of freedom). Of course, this reduction in significance may also be the result of having 20% fewer observations. This result suggests that in practice the largest benefits may come from the most colorable categories, but that algorithmic assignments are still useful (albeit less so) for less colorable categories. Moreover, we find no evidence in favor of ordered assignment over semantically-resonant algorithmic assignment.

6. Discussion and Future Work

We presented an algorithm for assigning semantically-resonant colors to categorical values. The resulting color assignments agree well with crowdsourced human-created color assignments. We also assessed the benefits of semantically-resonant color mappings on chart reading time via two controlled experiments. We find that both expert and algorithmic assignments significantly improve performance over default assignments, possibly through reduced reliance on chart legends. In addition, these benefits hold for a wider, more diverse set of categorical value sets. Our semantically-resonant color assignment algorithm is available as open-source software at http://github.com/StanfordHCI/semantic-colors.

We now conclude with a discussion of the implications of our work and possible directions for future work.

Significance of response times. As Gray and Boehm-Davis note [GBD00], "Whether designers intend to engineer interactions at the millisecond level, they do." While shaving off 0.5-1 seconds on a graph-reading task may seem small, this represents more than 10% of the task-time. Furthermore, analysts make many such comparisons, which add up.

In this paper, we investigated tasks that involve comparing quantities for associated categorical values. Additional experiments might be conducted to investigate the reverse task of finding values that correspond to salient quantities (e.g., name the fruit with the highest sales). Our crowdsourced experiments show that semantically-resonant colors can lower response times, but do not establish the cause(s) of this improvement. More controlled laboratory experiments might help identify these causes. For example, eye-tracking studies could be used to assess our hypothesis that semantically-resonant colors lead to reduced legend lookups.

Intelligent image query techniques. We used a simple query-expansion technique (adding "clipart"), and introduced a method to weight each expansion based on its relevance to the categorical value. An extension of this method may be used to incorporate more diverse data sources. For instance, image search may not provide adequate information, particularly for more abstract values (e.g., images for "angry" often depict facial expressions rather than a symbolic color). In these cases, using a knowledge base of concept-color associations akin to Havasi et al. [HSH10], and using semantic links to find colors through different levels of indirection, may be more useful. Query-expansion could also leverage information from other categorical values, or from the name of the category itself. For example, the value "apple" can refer to both the fruit and the computer company, and the name of the category ("fruit") or other categorical values ("cherries") may help disambiguate.

Balancing semantic resonance with perceptual constraints. This paper demonstrates benefits for semantically-resonant color assignments in visualization. However, there are other concerns, such as color discriminability, that arise when choosing visualization colors. In the present work, we address these concerns by using a palette (Tableau 20) specifically designed to satisfy such constraints. However, the general problem of balancing semantic resonance with perceptual constraints is still largely unsolved. Our current fixed-palette approach has shortcomings when categorical values have strong but perceptually similar color associations (e.g., "wind", "tornado", and "flood"). Here, color assignments from our algorithm tend to contain a narrow range of hues. Further experiments might explore tradeoffs between semantic resonance and discriminability to inform better automatic color assignments.

Palette design for algorithmic assignment. Our algorithm benefits from a predefined palette that helps ensure basic perceptual concerns are met. However, the Tableau 20 palette was designed assuming primarily arbitrary color assignments to values. Could palettes be designed to support algorithmic assignments from the ground up? Larger palettes with more hues (and perhaps color co-occurrence constraints) might extend the effectiveness of our approach. For example, such palettes may allow algorithms to select hues for a wider range of categories or use additional saturation or luminance levels when more resolution is needed (e.g., needing three different shades of green rather than two). Crafting new palettes to support semantically-resonant algorithmic assignment – while maintaining perceptual and aesthetic qualities – poses an interesting design challenge.

Acknowledgements

This work was supported by NSF Grant IIS-1017745.

References

[Baj88] BAJO M.: Semantic facilitation with pictures and words. *Journal of Experimental Psychology: Learning, Memory, and Cognition 14*, 4 (1988), 579. 1

[Ber91] BERRY L.: The interaction of color realism and pictorial recall memory. In *Proc. Association for Educational Communications and Technology* (1991). 1

[BRT95] BERGMAN L., ROGOWITZ B., TREINISH L.: A rule-based tool for assisting colormap selection. In *Proceedings of the 6th conference on Visualization'95* (1995), IEEE Computer Society, p. 118. 2

[BTBV02] BENAVENTE R., TOUS F., BALDRICH R., VANRELL M.: Statistical modelling of a colour naming space. In *Proceedings of the 1st European Conference on Color in Graphics, Imaging, and Vision (CGIV^{3}2002)* (2002), pp. 406–411. 2

[CM*84] CLEVELAND W., MCGILL R., ET AL.: Graphical perception and graphical methods for analyzing scientific data. *Journal of the American Statistical Association* (1984), 531–554. 6

[CSH08] CHUANG J., STONE M., HANRAHAN P.: A probabilistic model of the categorical association between colors. In *Color Imaging Conference* (2008), pp. 6–11. 2, 3

[CWM09] CHUANG J., WEISKOPF D., MÖLLER T.: Energy aware color sets. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 203–211. 2

[DA72] DALRYMPLE-ALFORD E.: Associative facilitation and interference in the Stroop color-word task. *Perception & Psychophysics 11* (1972), 274–276. 2

[GBD00] GRAY W., BOEHM-DAVIS D.: Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied 6*, 4 (2000), 322. 9

[HB03] HARROWER M., BREWER C.: Colorbrewer. org: An online tool for selecting colour schemes for maps. *Cartographic Journal, The 40*, 1 (2003), 27–37. 2, 4

[HB10] HEER J., BOSTOCK M.: Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design. In *ACM Human Factors in Computing Systems (CHI)* (2010), pp. 203–212. 3

[Hea96] HEALEY C.: Choosing effective colours for data visualization. In *Visualization'96. Proceedings.* (1996), IEEE, pp. 263–270. 2

[HS12] HEER J., STONE M.: Color naming models for color selection, image editing and palette design. In *ACM Human Factors in Computing Systems (CHI)* (2012). 2, 3, 4, 5

[HSH10] HAVASI C., SPEER R., HOLMGREN J.: Automated color selection using semantic knowledge. *Proceedings of AAAI CSK, Arlington, USA* (2010). 2, 9

[Kuh55] KUHN H.: The Hungarian method for the assignment problem. *Naval research logistics quarterly 2* (1955), 83–97. 4

[KZ10] KOSARA R., ZIEMKIEWICZ C.: Do Mechanical Turks dream of square pie charts? In *Proceedings of the 3rd BELIV'10 Workshop: BEyond time and errors: novel evaLuation methods for Information Visualization* (2010), ACM, pp. 63–70. 3

[LSS12] LEE S., SIPS M., SEIDEL H.: Perceptually-driven visibility optimization for categorical data visualization. *IEEE Transactions on Visualization and Computer Graphics* (2012). 2

[Mac91] MACLEOD C.: Half a century of research on the Stroop effect: An integrative review. *Psychological bulletin 109*, 2 (1991), 163. 2

[MMW10] MYLONAS D., MACDONALD L., WUERGER S.: Towards an online color naming model. In *Color Imaging Conference* (2010), pp. 140–144. 2, 3

[Mor03] MORONEY N.: Unconstrained web-based color naming experiment. In *Proc. SPIE* (2003), vol. 5008, pp. 36–46. 2, 3

[MSK04] MEIER B. J., SPALTER A. M., KARELITZ D. B.: Interactive color palette tools. *IEEE Comput. Graph. Appl. 24*, 3 (May 2004), 64–72. 2

[OAH11] O'DONOVAN P., AGARWALA A., HERTZMANN A.: Color compatibility from large datasets. In *ACM SIGGRAPH 2011 papers* (2011), SIGGRAPH '11, pp. 63:1–63:12. 2

[RDD00] ROBERSON D., DAVIES I., DAVIDOFF J.: Color categories are not universal: Replications and new evidence from a stone-age culture. *Journal of Experimental Psychology: General; Journal of Experimental Psychology: General 129*, 3 (2000), 369. 2

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: GrabCut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 309–314. 3

[RT90] RHEINGANS P., TEBBS B.: A tool for dynamic explorations of color mappings. In *ACM SIGGRAPH Computer Graphics* (1990), vol. 24, ACM, pp. 145–146. 2

[Sha02] SHARMA G.: *Digital color imaging handbook*, vol. 11. CRC, 2002. 3

[SKC*11] SAVVA M., KONG N., CHHAJTA A., FEI-FEI L., AGRAWALA M., HEER J.: ReVision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), ACM, pp. 393–402. 5, 8

[SL06] SPENCE I., LEWANDOWSKY S.: Displaying proportions and percentages. *Applied Cognitive Psychology 5*, 1 (2006), 61–77. 6

[SS12] SCHAUERTE B., STIEFELHAGEN R.: Learning robust color name models from web images. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)* (Tsukuba, Japan, November 11-15 2012), IEEE. 2, 3

[Sto08] STONE M.: Color in information display. In *IEEE Visualization 2008* (2008). 2

[SWD05] SHARMA G., WU W., DALAL E.: The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application 30*, 1 (2005), 21–30. 5

[TGH12] TALBOT J., GERTH J., HANRAHAN P.: An empirical model of slope ratio comparisons. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2012). 3

[Tuf90] TUFTE E.: *Envisioning information.* Graphics Press, 1990. 2

[VDWSVL09] VAN DE WEIJER J., SCHMID C., VERBEEK J., LARLUS D.: Learning color names for real-world applications. *Image Processing, IEEE Transactions on 18*, 7 (2009), 1512–1523. 2, 3

[WB07] WINAWER J. W. N. F. M. W. L. W. A., BORODITSKY L.: Russian blues reveal effects of language on color discrimination. *Proceedings of the National Academy of Sciences 104*, 19 (2007), 7780–7785. 2

[WGM*08] WANG L., GIESEN J., MCDONNELL K., ZOLLIKER P., MUELLER K.: Color design for illustrative visualization. *Visualization and Computer Graphics, IEEE Transactions on 14*, 6 (2008), 1739–1754. 2