Modeling How People Extract Color Themes from Images

Sharon Lin
Computer Science Department
Stanford University
sharonl@cs.stanford.edu

Pat Hanrahan
Computer Science Department
Stanford University
hanrahan@cs.stanford.edu

ABSTRACT
Color choice plays an important role in works of graphic art and design. However, it can be difficult to choose a compelling set of colors, or *color theme*, from scratch. In this work, we present a method for extracting color themes from images using a regression model trained on themes created by people. We collect 1600 themes from Mechanical Turk as well as from artists. We find that themes extracted by Turk participants were similar to ones extracted by artists. In addition, people tended to select diverse colors and focus on colors in salient image regions. We show that our model can match human-extracted themes more closely compared to previous work. Themes extracted by our model were also rated higher as representing the image than previous approaches in a Mechanical Turk study.

Author Keywords
color theme extraction; color themes; color names; algorithms; crowdsourcing

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

General Terms
Algorithms; Human Factors; Design; Measurement.

INTRODUCTION

Color choice plays an important role in setting the mood and character of a work of art and design. However, it can be difficult to choose good color combinations from scratch. Instead, artists, both expert and beginner, often draw colors from other sources of inspiration. These include other images and premade sets of color combinations called *color themes*.

There are many online communities, including Adobe Kuler [15] and COLOURlovers [4], that are centered around sharing and creating color themes. Many of these color themes are also created from images, rather than from scratch. Around 30% of a sampling of the newest 1,000 themes created on Colourlovers were created using their From-A-Photo theme tool.

CHI 2013, April 27–May 2, 2013, Paris, France.

In this work, we focus on color themes extracted from images. We consider the color theme of an image to be a small set of colors, usually 3 to 7, that best represent that image. Being able to automatically extract good image-associated themes can facilitate applications such as color picking interfaces [18] and color mood transfer from one image to another [10, 22]. Identifying the key colors in an image can also be useful in matching colors in a document or website around an image [20].

To our knowledge, this work is the first to evaluate and model color theme extraction based on the themes people pick from images. Previous work on automatically extracting color themes from images include general clustering techniques like k-means [16, 23] and fuzzy c-means [2] that focus on optimizing image recoloring error. We show that people often pick different colors than these algorithms. Other techniques include extracting colors successively from peaks in the image's color histogram [5, 6]. However, such a tiered approach can make it difficult to control the number of colors in the final theme. More recently, O'Donovan et al. [21] introduce a model to predict highly aesthetic themes by training on large online theme datasets. They consider themes in the general context, while we look specifically at themes extracted from images.

This work has two main contributions. First, we a present a method to evaluate theme extraction techniques against human-extracted themes using theme *overlap* and theme *distance*. Second, we introduce a regression model trained on a corpus of human-extracted themes and their associated source images. The fitted model can then be used to extract color themes from other images. We show that our model extracts themes that match human-extracted themes more closely than previous approaches. Online study participants also rate the model-extracted themes higher as representing the source image than themes extracted by k-means and an aesthetics-based approach.

RELATED WORK

Previous approaches have proposed quantitative measures for evaluating the quality of a theme based on either recoloring error [23, 2], aesthetics [21], or color nameability [12]. However, to our knowledge, this is the first approach that compares image-based color themes to ones that people have manually extracted.

Updated August 30, 2013: Fixed code bug for evaluating agreement between the same set of themes. Artist-extracted themes are not as noticeably more consistent than Turk-extracted themes.

Clustering and Histogram-based Approaches

One common method for extracting a representative set of colors is to use general clustering techniques, such as k-means [16, 23] and fuzzy c-means clustering [2]. K-means takes a number of requested colors k, and attempts to find clusters that minimize recoloring error. It does not take into account spatial arrangement of the colors in the image, and can thus wash out important image regions. Fuzzy c-means clustering is similar to k-means, except with soft instead of hard assignment of pixels to clusters, and so it is less affected by outliers. These approaches evaluate color themes based on a quantitative metric: the recoloring error. However, this may not be the only metric people use to evaluate themes.

Delon et al. [5, 6] found that peaks in the image's color histogram often correspond to spatial regions in natural imagery. Their algorithm extracts color themes by successively finding meaningful peaks in the Hue, Saturation, and Value histograms of the image. The resulting color set often contains many colors, some of them redundant, due to the tiered extraction approach. Morse et al. [19] used a similar tiered histogram approach to extract color themes given user-specified constraints on the maximum number of colors and a minimum distance between colors. However, they provided no user or quantitative evaluation of the themes against other approaches.

Color Harmony and Theme Aesthetics

Many online color theme creators allow users to design themes based on popular harmony templates [13, 17], predefined relationships of colors on the hue wheel. These relationships are often used as guidelines when creating themes from scratch. O'Donovan et al. [21] investigated the impact of color harmony templates on themes within large-scale online theme datasets. They found little evidence that people naturally gravitated towards harmony templates or that following these templates increased aesthetic ratings.

Our method uses a similar data-driven approach as O'Donovan et al., who predict the aesthetic rating of a color theme using a regression model trained on online theme datasets. Their model considered low-level features such as the values of each color component in the theme in multiple color spaces and differences between adjacent colors.

However, O'Donovan et al. focused on color themes and their ratings without context of where the theme originated. This paper looks more specifically at color themes that are paired with images. Instead of modeling themes with high aesthetic ratings, we look at the problem of characterizing themes that best capture an image, which itself may be aesthetically pleasing.

Color Names

Previous research in color names [1, 3, 12] has developed models and corresponding metrics for categorical color perception. Color names, such as *red* and *light blue*, are the descriptions people use to communicate a color. Chuang et al. [3] introduced a probabilistic model for these color-name associations, learned from the 330 colors in the World Color Survey [1]. Colors that are more consistently and uniquely named are considered to have higher saliency.

More recently, Heer and Stone [12] built upon this probabilistic model and trained on a much larger XKCD online color name survey. They also defined a distance metric between two colors as the distance between the associated name distributions. Heer and Stone looked at color name features for assessing themes used in data visualization applications. They hypothesized that using colors with unique names would make it easier for people to verbally communicate different elements in the visualization than when using colors with overlapping names. In this work, we look at these color nameability and color name difference features [12] as potential predictors for how people extract themes from images.

GATHERING THEMES FROM PEOPLE

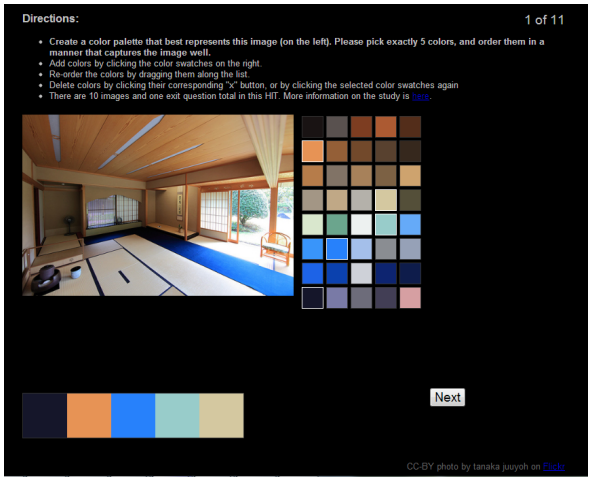

Figure 1. The user interface for the theme collection experiment with source image on the left, swatches on the right, and created theme below. Image credit: Tanaka Juuyoh (tanaka_juuyoh on Flickr)

To gather a dataset of human-extracted color themes, we asked people to extract themes from a set of 40 images. These images consisted of 20 paintings and 20 photographs. We varied the type of image to counter the effects of image style and content on the themes extracted. The paintings were chosen from five artists with different artistic styles (Impressionist, Expressionist, Pointillist, Realist, and Ukiyo-e prints). The photographs were Flickr Creative Commons images chosen from the categories Landscape, Architecture, Interior, Closeup, and Portrait.

We gathered themes from Amazon Mechanical Turk, which has been used successfully in crowdsourcing graphical perception [11] and creative sketching tasks [7]. One potential issue with crowdsourcing color themes is that we cannot easily control for different monitor and lighting conditions, which can introduce more noise in the collected data. However, in practice, people often view and create color themes under different conditions. Thus, by gathering themes from many different people, we can later fit a model that averages over typical viewing conditions rather than one that targets a specific condition.

Pilot studies determined that Turk participants often did not take the time to choose color shades carefully by clicking on the image directly. In addition, giving no limitation on the number of colors chosen resulted in color themes with wide variance in size. Therefore, we constrained the study design by requiring participants to choose exactly 5 colors from candidate color swatches. Color themes of size 5 have been studied previously [21] and are also the most common on online theme sharing sites.

For each image, we generated 40 color swatches by running k-means clustering on the image. The initial seeds for the clustering were stratified randomly sampled within the CIELAB bounding box of the image. The resulting swatch colors were snapped to the nearest pixel color in the image.

We asked participants to extract themes from either 10 paintings or 10 photographs. Participants were shown one image at a time and its associated color swatches. They were asked to pick 5 different colors that would "best represent the image and order them in a way that would capture the image well." The interface allowed for participants to add, remove, and reorder color swatches in their created theme. The order of images was counter-balanced using a balanced Latin square design. In total, we recruited 160 different participants and collected a total of 1600 themes (40 themes per image). Each Turk task was \$0.50 (\$0.05 per theme) and was limited to participants in the United States. The median time to complete one theme was 24 seconds. All images and color swatches were shown on a black background to match previous color theme rating studies [21] and popular online theme creation tools. At the end of the study, participants were asked to describe their strategy for choosing which colors to include in their themes.

For comparison purposes, we also asked 11 art students to extract themes from a randomly chosen subset of 10 images (5 paintings and 5 photographs). The interface for the art students was the same as for the Mechanical Turk participants, and image order was randomized within the paintings and the photographs. Art student participants were compensated with a \$5 gift card after the study. For art students, the median time to complete one theme was 20 seconds.

Theme-Gathering Results

Figure 2 shows all the swatches presented to participants for one image, and each human-extracted theme as a column to the right of the swatches. The themes chosen by k-means and c-means clustering with k set to 5 is shown on the left of the swatches. Qualitatively, people agree with each other on certain key colors, shown by the strong horizontal lines in the figure, with some variability in the exact shade. K-means and c-means clustering often fail to select the common colors chosen by people.

In order to compare the consistency of participants quantitatively, we look at the mean *overlap* (number of colors in common) between all pairs of collected themes. We first match up the colors in one theme to the other to achieve the minimum total error, the minimum bipartite matching. The overlap is the number of color matchings that fall below a given distance threshold:

$$overlap(A, B, t) = \sum_{(a,b)\in m(A,B)} [\|a - b\|_2 < t] \quad (1)$$

where A and B are themes, $m(A, B)$ is the minimum bipartite matching, and t is the distance threshold.

Figure 3 plots the average overlap between themes from different sources against the distance threshold. Colors from k-means and c-means are snapped to the nearest candidate swatch color in the graph. For low distance thresholds (e.g. 0), colors from these methods would never overlap with colors chosen from the swatches by people. This snapping gives the algorithms which operate on continuous color space a fair footing when comparing them against choices made by participants.

On average, people agreed on nearly 2 out of 5 color swatches per theme. Mechanical Turk participants and artists roughly agreed on particular color shades. In comparison, random, c-means, and k-means themes all agreed poorly with human-extracted themes when considering particular color shades.

TRAINING A MODEL OF THEME-EXTRACTION

Given the dataset of images and their associated themes, we train a model for characterizing a human-extracted theme. Our basic approach is to first compute target scores for each theme on how close it is to human-extracted themes, generate many themes with different scores, and then calculate features describing them. Finally we use LASSO regression [9] to fit a linear model to predict the target scores given the theme features. Once fitted, this model can later be used to extract themes from images without human-extracted theme data.

Theme Similarity to Human-Extracted Themes

We define the *distance* between two themes to be the minimum total error from a bipartite matching of each color in one theme to a color in the other theme. The score for how similar a theme is to human-extracted themes is then the average distance between that theme and all human-extracted themes. This can be expressed as:

$$score(p) = 1 - \frac{1}{|H|} \sum_{h\in H} \frac{dist(p, h)}{maxDist} \quad (2)$$

where p is the given theme in question, H is the set of human-extracted themes, $dist$ is the total Euclidean error between the two themes in CIELAB color space, and $maxDist$ is some maximum possible distance between two themes. The theme scores are then rescaled between 0 and 1 for each image, so that each image gets equal weight in training. Themes with scores closer to 1 are more perceptually similar to human themes on average than themes with scores closer to 0.

We find that the top 30 (75%) representative Turk-extracted themes out of 40 for each image agree more closely with the artist-extracted themes, about as closely as artist-extracted

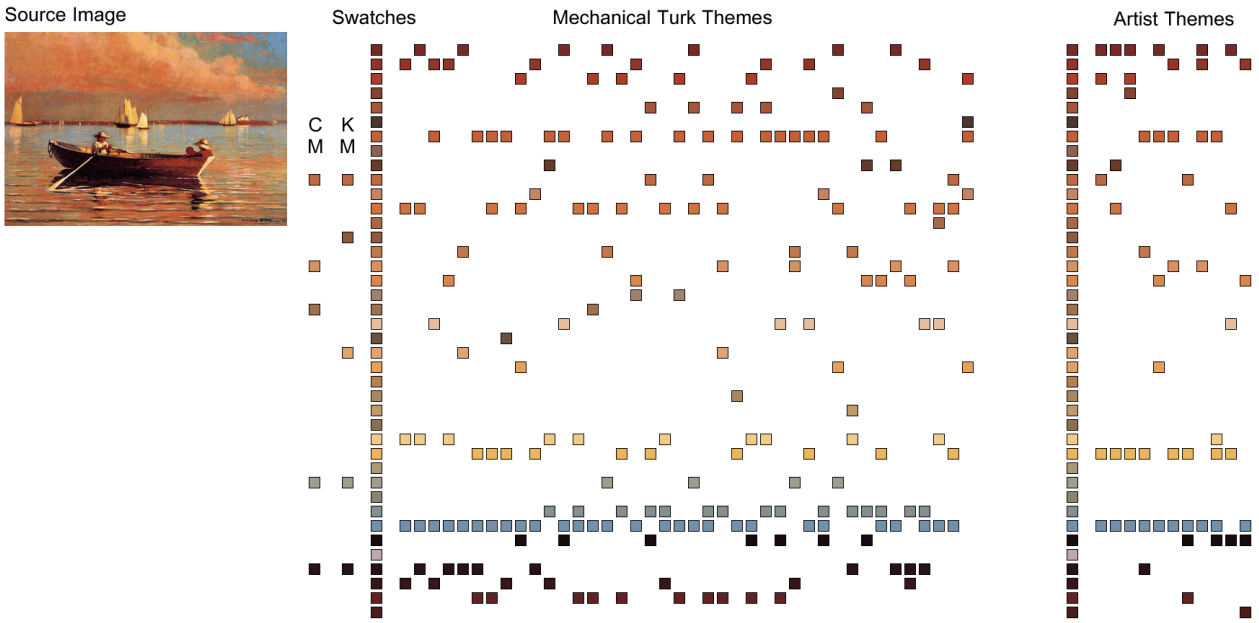

Figure 2. All the color themes for the source image. The color swatch options are shown down the middle. The human-extracted themes are on the right, with each column being a separate theme. The themes chosen by k-means (KM) and c-means (CM) are shown on the left. Image credit: Homer

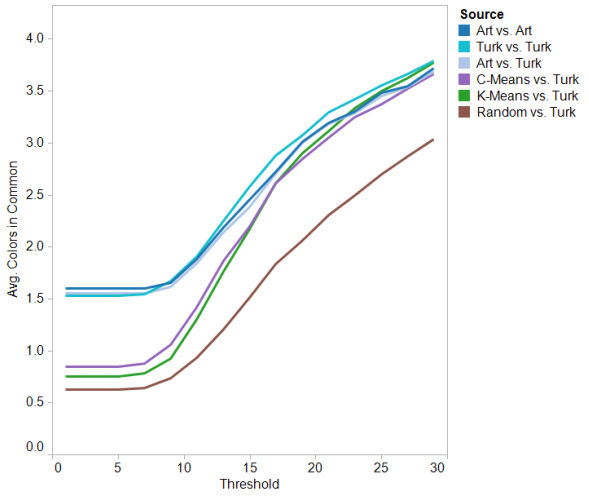

Figure 3. The average number of colors in common between themes from different sources plotted over a distance threshold. Blue lines are humans compared to humans. Other colored lines are automatic algorithms compared to humans.

Figure 4. Examples of oracle color themes, which have closest distance on average to the Turk-extracted color themes for each image. Image credits: Per Ola Wiberg (powi on Flickr); Seurat

themes agree with each other. For the rest of the analyses in the paper, we therefore remove the bottom 10 themes from each image (25%) as outliers.

Given this distance metric and the human-extracted themes for an image, we can find an optimal *oracle* color theme that is closest on average to all the human-extracted themes. This provides us with a way to sanity check our distance metric as well as provide a theoretical upper bound of performance for automatic algorithms. Figure 4 shows the oracle color themes for two example images.

The oracle themes were created by hill-climbing over the candidate color swatches shown to participants in the palette-extraction experiment. In this method, we pick a random starting theme of 5 different colors from the candidate swatches. Then for each color in the theme, we find a replacement color from the candidate swatches that would most increase the score. We repeat this process until no further replacements can be made to increase the score. This method will find local, though not necessarily global, optima. Thus, we re-run hill-climbing for several (in this case 10) random restarts and pick the result with the best score.

LASSO Regression

We randomly generate 1000 themes per image with scores evenly distributed among 10 bins between 0 and 1. The 10 images shown in the artist experiment and their associated themes are reserved as a test set. The rest of the themes are used for training.

We use LASSO regression to fit a linear model to the training set. LASSO regression attempts to model the theme score in Equation 2 as a weighted sum of features and an intercept $b + \sum_i w_i \cdot f_i$. It also does feature selection by penalizing potential models by the L1 norm of their feature weights. This means that LASSO will find a model that both predicts the target scores well and also does not contain too many features. For each theme, we calculate a total of 79 features and use LASSO to find the features most predictive of human-extracted themes. The hyper-parameter λ determines the sparsity of the model and was tuned to minimize 10-fold cross-validation error in the training set (with 3 images and their associated themes in each fold).

In this work, we consider six types of features to describe each theme: saliency, coverage error both for pixels and for segments, color diversity, color impurity, color nameability, and cluster statistics. Within each type of feature, we calculate several variations using different distance metrics and parameters. Several of the features are highlighted below.

Saliency

Most study participants reported that they picked colors which "popped out of the image", "caught their eye", or were "the most salient colors." To detect salient regions in the image, we compute image saliency maps according to the work of Judd et. al. [14], who learned a model of saliency from eye tracking data on natural photographs. These maps were computed taking into account both low-level features and semantic features such as horizon lines and faces. They assign a saliency value to each pixel in the image.

We assign each image pixel to the nearest candidate color swatch shown to participants. The saliency of a color swatch is the sum of its individual pixel saliencies. The *total saliency* captured by a theme, $sal(C)$, is then the sum of its color swatch saliencies, relative to the maximum capturable saliency. Formally,

$$sal(C) = \frac{1}{max} \sum_{c \in C} \sum_{p \in cluster(c)} saliency(p) \qquad (3)$$

where C is the set of five swatches in the theme, $cluster(c)$ is the set of pixels quantized to swatch c, and max is the total saliency of the top 5 most salient swatches.

In addition to the total saliency, we also look at min, max, and average *salient density* of the colors in the theme. The salient density of a color, $sd(c)$, is calculated as the saliency of the color swatch divided by the number of pixels assigned to that swatch. Cluster assignments can be made among the candidate color swatches or the theme colors.

$$sd(c) = \frac{1}{|cluster(c)|} \sum_{p \in cluster(c)} saliency(p) \qquad (4)$$

Pixel Coverage

One feature people may take into account when choosing theme colors is how well the colors cover the overall image. We consider two metrics: *recoloring error* and color channel *range coverage*.

Recoloring error is defined as the total error resulting from recoloring each pixel in the image with the theme colors. We define *hard* recoloring error as:

$$hcov(C, I) = \sum_{p \in I} w_p \cdot \min_{c \in C} error(p, c) \qquad (5)$$

where I is the set of pixels in the image, w_p is the weight of pixel p, and c is a theme color. Intuitively, this is the error resulting from recoloring each pixel with the closest theme color. K-means clustering minimizes a variant of this feature with uniform pixel weights and squared Euclidean distance as the error function.

We replace the error function with Euclidean distance and squared Euclidean distance in a perceptually-based color space (CIELAB) and color name cosine distance [12]. Distances are normalized according to the maximum color swatch distance. In addition, we either weight each pixel uniformly with $w_p = \frac{1}{size(I)}$, or we weight each pixel according to their saliency in the image.

We also define *soft* recoloring error as:

$$scov(C, I) = \sum_{p \in I} w_p \cdot \sum_{c \in C} u_{pc}^2 \cdot error(p, c)^2 \qquad (6)$$

$$u_{pc} = \frac{1}{\sum_{j \in C} \left(\frac{error(p,c)}{error(p,j)} \right)^2} \qquad (7)$$

where each pixel can take different recoloring contributions from each theme color. This is the objective function that fuzzy c-means clustering attempts to minimize. Again, we vary the error function with Euclidean distance in CIELAB space and color name cosine distance.

In addition, we consider the lightness (L), red-green (A), and blue-yellow (B) range of the image compared to the range of the theme in CIELAB space. Saturation (S) range coverage in HSV space is also considered. For lightness coverage:

$$Lcov(C) = \frac{range(C)}{range(I)} \qquad (8)$$

where $range(I)$ is the difference between the maximum and minimum L values in the image swatches, and $range(C)$ is the difference for the theme. Red-green, blue-yellow, and saturation coverage are defined similarly.

Segment Coverage

People interpret images as arrangements of objects and components instead of on a pixel-level scale. Thus, we also include features that consider segments instead of just pixels. We segment the images using the method of Felzenszwalb and Huttenlocher [8].

The first feature is segment recoloring error, which is a weighted sum of the average recoloring error within each segment. *Hard* segment recoloring error is defined as:

$$hsegcov(C) = \sum_{s \in S} w_s \cdot hcov(C, s) \quad (9)$$

Similarly, *soft* segment recoloring error is:

$$ssegcov(C) = \sum_{s \in S} w_s \cdot scov(C, s) \quad (10)$$

with pixel weights $w_i = \frac{1}{size(s)}$. S is the set of segments. The segment weights w_s can be either uniform or based on the relative saliency or salient density of the segment in the image.

Secondly, we also consider the *uniqueness of the segment color* among the theme colors, $uniq(C)$. The idea is that colors in a theme may be evenly distributed among segments, so that no one segment would be sourced from most of the theme colors. To model this, we calculate the mean negative entropy of segments being colored by a particular theme color.

$$uniq(C) = \sum_{s \in S} w_s \cdot \sum_{c \in C} p(c|s) \ln p(c|s) \quad (11)$$

where $p(c|s)$ is the probability of a segment s being colored by c from the theme.

For each segment in the image, we calculate the distances from its mean color to the colors in the given theme. The probability of a segment taking on a given color from the theme is then its relative distance to that color compared to all other colors in the theme.

Color Diversity

We calculate several metrics for *color diversity*. These include the mean distance between one color and its closest color in the theme and the min, max, and mean distance between two colors in the theme.

Similarly, we use either CIELAB or color names as the distance metric. We normalize the distances by either the max or mean distance between the candidate color swatches shown to the user.

Color Impurity

The *impurity* of a theme color is computed as the mean distance between the theme color and its n% closest pixels in the image. O'Donovan et al. used this metric when applying their aesthetics model to extracting color themes from images [21]. Following their work, we chose n to be 5%.

We normalize distances by either the max or the mean distance between the candidate color swatches.

Color Nameability

In data visualization, one desirable trait for a theme may be how easy it is to refer to a color in the legend [12]. Similarly, for general images, people may extract the most characteristic color shades for a particular color category.

We compute the *nameability* of colors used in the themes and normalize by either the max or mean nameability in the candidate color swatches. Color nameability used here is the same as the color saliency metric used by Heer and Stone [12], but rescaled to the nameability range of the candidate color swatches. It describes how consistently and uniquely a given color is named.

Cluster Statistics

After quantizing image pixels to theme colors, we compute variance statistics to describe the resulting clusters. We look at the average *within-cluster variance* of image pixels around each theme color. The *between variance* is just the variance of the theme colors around the mean theme color.

RESULTS

Predictive Features of Human-Extracted Color Themes

Relative weights in the fitted model can indicate which sets of features predict human-extracted color themes well. Features with large weights create one set of good predictors. Features with small or zero weights tend to be uninformative or are redundant with these features. In our model, 40 of the 79 features were given non-zero weights. These weights are listed in the Appendix. For this analysis, we standardize the weights to better compare them across features.

Weighted soft recoloring error and color diversity features consistently have the largest weights in our model. Themes that contain the right color for salient regions in the image and have a variety of colors tend to be closer to human-extracted themes. Other weighted features included saturation range coverage, color impurity, and segment color uniqueness. Good themes tended to cover the range of saturations in the image well. In addition, themes that contained good color clusters in the image and did not focus too many colors on one image region were also boosted. Color nameability had small negative weights, possibly because highly nameable colors may be less used in photographs and paintings and also less aesthetically pleasing.

A remaining question pertains to the stability of these weights as the number of training images varies. Although the exact weights of the metrics shift as the number of training images grows, the top feature types in the model tends to stay the same. For example, the soft recoloring error per segment and color diversity remain the highest-weighted features as we increase the number of training images from 10 to 30 for constant lambda. In addition, the change in weights decreases as the number of training images grows to 30. Thus, we believe 30 images is a reasonable training set size, though more images could help stabilize the weights further.

One important note is that while LASSO regression selects a set of features that fits the training data well, there may be other feature sets with similar predictive power. Further investigation is needed to explore the tradeoffs between models with different feature sets and performance.

Matching Human-Extracted Themes

On our test set of images, the mean absolute error (MAE) from running the fitted linear model was 0.10 compared to

the 0.22 of a fixed baseline for the target scores. We use the fitted model to extract color themes from the test set of 10 images by hill-climbing over the candidate color swatches. This is identical to our approach when finding the oracle themes, except we use the model to predict the scores instead of the actual human-extracted themes.

Figure 5 plots themes created using our model, k-means, c-means, and random selection against artist-created themes on the test set of 10 images. We also plot the Turk oracle themes against the artist-created themes to see the theoretical maximum agreement. For the graph, we again snap the colors in the themes to the closest swatch color shown to the human participants.

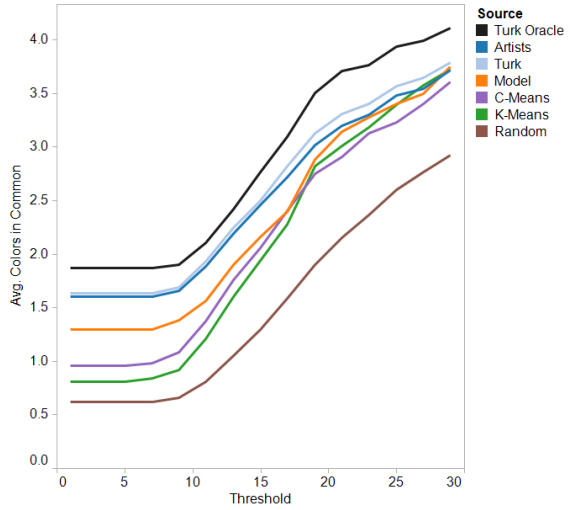

Figure 5. Theme agreement (number of colors in common) compared to artist-extracted themes on the test set of 10 images as color distance threshold increases.

Reference	Artists	Turk	Model	CM	KM	Rand
Artists	20.6	19.6	**20.4**	22.9	22.7	29.1
Turk	19.6	16.8	**18.6**	21.3	20.3	28.4

Table 1. Average distances per color between color themes of different methods compared to humans. Units are in CIELAB color space. Abbreviations are our model (Model), k-means (KM), c-means(CM), and Random (Rand)

The oracle themes from Turk agreed closely with the artist-extracted themes overall, moreso than themes from the average Turk or artist participant. This indicates that if we are able to perfectly model our optimization function, we can extract good color themes.

Our model-extracted themes agreed more closely with artist-extracted themes than do themes from other algorithms. In addition, the average distance of the human-extracted themes to the model-extracted themes is smaller than for the other algorithms, shown in Table 1. Reported distances are given for the original colors, not ones snapped to the color swatches.

For evaluation with previous work, we gathered human-extracted color themes for the 40 images used by O'Donovan et al. [21]. Figure 6 shows the similarity of themes extracted from different algorithms to human-extracted themes from Mechanical Turk. The aesthetics-enhanced model (OD-Aesthetic) [21] performed slightly better than the original without the aesthetics term (OD-Original), which indicates that aesthetics may play a role in the colors people choose. In this second test set, our model again matched human-extracted themes from Turk more closely than the other algorithms, shown in Table 2.

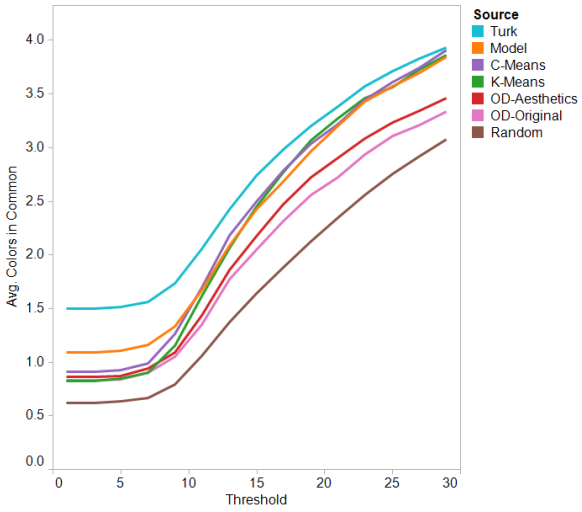

Figure 6. Theme agreement (number of colors in common) compared to Turk-extracted themes on the larger test set of 40 images as color distance threshold increases

Reference	Turk	Model	OD [21]	CM	KM	Rand
Turk	18.3	**20.6**	26.1	21.0	22.0	28.3

Table 2. Average distances per color between color themes of different methods compared to humans on a larger test set of 40 images. Units are in CIELAB color space. Abbreviations are our model (Model), the aesthetics-enhanced model by O'Donovan et al. (OD), k-means (KM), c-means(CM), and Random (Rand)

Representing the Image

Quantitatively, our model-extracted themes closely match human-extracted themes for the test images. But how well do the model-extracted themes actually represent the color theme of the image?

To answer this question, we conducted a study on Mechanical Turk asking 40 participants to rate color themes for 20 random images from the O'Donovan test set. The task was limited to participants in the United States. Figure 7 shows the study interface. Participants were shown one image at a time and 4 associated color themes: a representative human-extracted theme (nearest to other human-extracted themes), our model-extracted theme, a k-means theme, and an aesthetics-based theme from O'Donovan et. al. [21]. They were asked to rate the color themes on "how well they represent the color theme of the image" on a Likert scale from 1 (Not well at all) to 5 (Very well). Theme order was randomized, and image order was counter-balanced using a Latin Square design. The order of colors in the model-extracted and k-means themes was determined by their CIELAB distance to red. Each participant was paid $1.

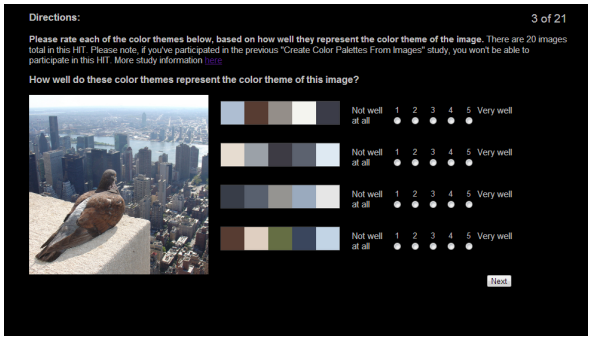

Figure 7. The interface for the theme rating study with source image on the left and themes on the right. Participants were asked to rate each theme on how well it represents the color theme of the image. Image credit: ZeroOne (villes on Flickr)

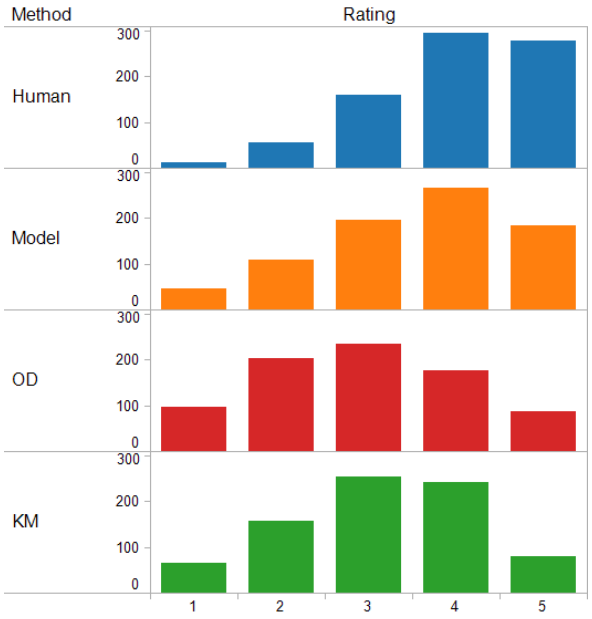

Figure 8. Histogram of theme ratings for human-extracted, our model, O'Donovan et al.(OD), and k-means themes

Figure 8 shows the distribution of ratings for each method according to how well their themes represented the color theme of the image. Overall, human-extracted themes were rated highest (Mean: 3.97), then our model-extracted themes (Mean: 3.54), k-means themes (Mean: 3.14), and the aesthetics-enhanced themes (Mean: 2.94). This indicates a correlation between how closely a theme matches human-extracted themes and how well it is rated as representing the image.

We ran a repeated measures ANOVA on the ratings with the method as a fixed effect and participant and image as random effects. There was a significant effect of the methods on the ratings ($p < 0.001$). We then ran follow-up paired t-tests using Bonferroni correction for each pair of methods. Each image and participant combination was treated as a repeated observation of the method. The differences between the mean ratings for each method were all significant at $p < 0.001$.

Figure 9. Examples of images and their associated themes from people (H), our model (M), k-means (K), and an aesthetics-based model in O'Donovan et al. 2011 (OD). H, M, and K themes are re-aligned to match the OD themes for easier comparison. Image credits: Turner; Monet; Ajith U (uajith_set1 on Flickr); Mike Behnken (mikebehnken on Flickr)

It should be noted that the experiment tested how well a theme captures the color theme of the image, and not how generally aesthetically pleasing the theme is. The results show that themes which best represent an image and themes that are optimized for general aesthetics may be different.

Figure 9 shows examples of the 4 different themes shown to participants for 5 images. Our model tends to extract vivid and bold colors, which are often ones chosen by people, as it has learned that themes with large distances between colors are usually more fitting. However, the last image in the figure shows a case where our model extracts a very bold theme that includes bright green and red, which may not be desirable. Although people often chose these colors individually, they rarely included them together in a theme. This may be a byproduct of the training set of 30 images, where the distribution of image styles tended to be larger than in this test set of images, which focused more on photographs.

DISCUSSION

Themes from our model closely match human-extracted themes compared to other algorithms, though there is still room for improvement. More images and human-extracted

themes can help smooth out biases in the model. Improvements in object recognition, segmentation, and image saliency maps are also likely to help our model. For example, face detection used in the image saliency model [14] works well in photographs, but usually fails on stylized images. Moreover, additional knowledge about semantics and object hierarchy in the image may help prioritize colors for very colorful images. A more in depth notion of aesthetics or harmony may also be predictive of the color shades people pick. More complex models, such as specially-designed graphical models, may better capture situation-dependent choices made by people.

However, our framework is flexible and can accommodate larger sets of images and additional features as necessary. There are many people interested in art who are creating color themes from images online each day, and these themes could provide data from which to learn. A similar framework could perhaps be used to learn good color themes for more focused application scenarios, such as web design, interior design, and data visualization.

There are many potential applications for color themes paired with their associated images. It could provide a method for image search for images with similar color themes. Images also provide context for how a color theme can be used, and the two together can assist colorization of patterns or web elements to match a given image. Drawing and painting programs can also personalize color swatches based on the color themes of a user's collection of favorite images.

CONCLUSION AND FUTURE WORK

In this paper, we present a framework for evaluating automatic color theme extraction algorithms against themes people extract. We show that people choose colors that are different from the widely-used k-means and c-means clustering algorithms.

In addition, this work presents a first step in learning how to extract good color themes based on human-extracted theme data. We show that a linear model fitted on a training set of 30 images and their associated human-extracted themes outperforms many of the previous approaches. High-scoring themes tended to have diverse colors, focused on getting accurate colors for salient image regions, picked colors that are well-concentrated in the image, and spread colors evenly across image regions.

Future work includes looking at how people choose colors for more focused image classes, such as web design, visualization, or particular art styles. We could learn what features of color themes are most characteristic for each scenario, how they differ, and if there are any trends in color combinations.

Color themes are also only one component of how people interpret works of art and design. A similar data-driven approach could be used to learn important features for other graphical aspects, such as texture or shading. Increasing our understanding in these areas could perhaps enable better tools for assisting users in art and design tasks.

ACKNOWLEDGMENTS

We thank Peter O'Donovan for his help in comparing results. Thank you to Jeffrey Heer and Theresa-Marie Rhyne for helpful feedback, and to all our study participants. This work was funded by NSF FODAVA grant CCF-0937123.

REFERENCES

1. Berlin, B., and Kay, P. *Basic color terms: Their universality and evolution*. Univ of California Pr, 1991.
2. Bezdek, J. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.
3. Chuang, J., Stone, M., and Hanrahan, P. A probabilistic model of the categorical association between colors. In *Color Imaging Conference* (2008), 6–11.
4. Colourlovers. **http://www.colourlovers.com**.
5. Delon, J., Desolneux, A., Lisani, J., and Petro, A. Automatic color palette. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2 (sept. 2005), II – 706–9.
6. Delon, J., Desolneux, A., Lisani, J. L., and Petro, A. B. Automatic color palette. *Inverse Problems and Imaging 1*, 2 (2007), 265–287.
7. Eitz, M., Hays, J., and Alexa, M. How do humans sketch objects? *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 44.
8. Felzenszwalb, P. F., and Huttenlocher, D. P. Efficient graph-based image segmentation. *Int. J. Comput. Vision 59*, 2 (Sept. 2004), 167–181.
9. Friedman, J., Hastie, T., and Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software 33*, 1 (2010), 1.
10. Greenfield, G. R., and House, D. H. Image recoloring induced by palette color associations. *Journal of WSCG 11* (2003), 189–196.
11. Heer, J., and Bostock, M. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *ACM Human Factors in Computing Systems (CHI)* (2010), 203–212.
12. Heer, J., and Stone, M. Color naming models for color selection, image editing and palette design. In *ACM Human Factors in Computing Systems (CHI)* (2012).
13. Itten., J. The art of color. *Van Nostrand Reinhold Company* (1960).
14. Judd, T., Ehinger, K., Durant, F., and Torralba, A. Learning to predict where humans look. In *IEEE International Conference on Computer Vision (ICCV)* (2009).
15. Adobe Kuler. **http://kuler.adobe.com**.
16. MacQueen, J., et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, California, USA (1967), 14.
17. Matsuda, Y. Color design. *Asakura Shoten* (1995).
18. Meier, B. J., Spalter, A. M., and Karelitz, D. B. Interactive color palette tools. *IEEE Comput. Graph. Appl. 24*, 3 (May 2004), 64–72.
19. Morse, B., Thornton, D., Xia, Q., and Uibel, J. Image-based color schemes. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 3, IEEE (2007), III–497.
20. Obrador, P. Automatic color scheme picker for document templates based on image analysis and dual problem. In *Proceedings of SPIE*, vol. 6076 (2006), 64–73.
21. O'Donovan, P., Agarwala, A., and Hertzmann, A. Color compatibility from large datasets. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, ACM (New York, NY, USA, 2011), 63:1–63:12.
22. Wang, B., Yu, Y., Wong, T.-T., Chen, C., and Xu, Y.-Q. Data-driven image color theme enhancement. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA '10, ACM (New York, NY, USA, 2010), 146:1–146:10.
23. Weeks, A., and Hague, G. Color segmentation in the hsi color space using the k-means algorithm. In *Proceedings of SPIE*, vol. 3026 (1997), 143.

APPENDIX

Recoloring Error				
Components	Weighted By	Type	Metric	Weights
Image Pixels	Uniform	Hard	Dist	0
			SqDist	**0.9392**
			CN	0.2083
			SqCN	0
		Soft	Dist	0
			CN	0
	Saliency	Hard	Dist	0
			SqDist	0
			CN	0
			SqCN	0
		Soft	Dist	0
			CN	0
Segment Pixels	Uniform	Hard	Dist	0
			SqDist	**1.2941**
			CN	0.2439
			SqCN	0
		Soft	Dist	0
			CN	0
	Saliency	Hard	Dist	-0.0087
			SqDist	0
			CN	0
			SqCN	0
		Soft	Dist	**-7.1864**
			CN	**-2.0479**
	Salient Density	Hard	Dist	0
			SqDist	0
			CN	0
			SqCN	0
		Soft	Dist	**-9.9725**
			CN	0
Segment Mean	Uniform	Hard	Dist	0
			CN	0
	Saliency	Hard	Dist	0
			CN	-0.0496
	Salient Density	Hard	Dist	-0.0828
			CN	0

Range Coverage	
Type	Weights
Lightness (L)	-0.1909
Red-Green (A)	0.1601
Blue-Yellow (B)	0.0982
Saturation (S)	**0.8743**

Segment Uniqueness	
Weighted	Weights
Uniform	**0.7110**
Saliency	0

Diversity			
Space	Normalize	Metric	Weights
CIELAB	$max_D(I)$	min	0
		max	-0.0274
		mean	**1.5602**
		closest	0.3292
	$mean_D(I)$	min	0.0322
		max	0
		mean	0.1450
		closest	0.0299
CN	$max_D(I)$	min	0
		max	-0.0798
		mean	-0.0729
		closest	-0.0061
	$mean_D(I)$	min	0.0068
		max	0.0157
		mean	0
		closest	0

Nameability			
Space	Normalize	Metric	Weights
CN	$max_N(I)$	min	0
		max	-0.0411
		mean	0
	$mean_N(I)$	min	-0.0014
		max	0
		mean	0

Impurity			
Space	Normalize	Metric	Weights
CIELAB	$max_D(I)$	min	0.0278
		max	0
		mean	0
	$mean_D(I)$	min	0.0423
		max	-0.0355
		mean	**-0.6297**

Saliency		
Clusters	Metric	Weights
Swatches	min	-0.0408
	max	0
	mean	0.2937
	total	-0.1166
Theme	min	-0.0958
	max	0.1537
	mean	-0.1715

Cluster Statistics		
Type	Normalize	Weights
Within Variance	$max_D(I)^2$	-0.4973
Between Variance	$max_D(I)^2$	**-2.1370**

Table 3. All features and weights considered by the regression, organized by feature type and broken down by variations in parameters. Weights with magnitudes greater than 0.5 are highlighted. Abbreviations: CN - Color Name cosine distance, Sq - Squared, Dist - CIELAB Euclidean distance. Variations under Recoloring Error would be interpreted as recoloring error within Components:C, Weighted By:W, using Type:T assignments with the distance Metric:M. Similarly, Diversity variations would be interpreted as distances within the color Space:S, normalized by Normalize:N, using the Metric:M. Normalization terms can be either the mean or max distance or nameability between image swatches. Saliency variations are interpreted as using the Metric:M with saliency determined by clusters among the Clusters:C.