Automatic Color Palette Creation from Words

Albrecht Lindner and Sabine Süsstrunk
School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract

We present an automatic framework to extract color palettes from words. This is a novel approach in comparison to existing solutions, e.g. manual creation or extraction from images. The associations between words and colors are deduced from a large database of 6 million tagged images using a scalable data-mining technique. The palette creation can be constrained by the user to achieve a desired hue template. We first focus on single words and then extend to entire texts. We compare our results against Adobe Kuler, a widely used online platform of manually created color palettes. We show that our approach performs slightly better than its non-automatic counterpart in terms of user's preference rankings. This is a good result because our method is fully automatic whereas Kuler relies on users' palettes that are manually created and annotated.

Introduction

Color palettes are widely used by artists and designers to colorize documents, such as posters or webpages. The colors in palettes often compose a theme with an emotion that the artists desire to evoke in the observer. Traditionally, palettes are created manually or extracted from an input image. This paper proposes a novel technique to automatically create color palettes from words. This offers to artists the opportunity to freely verbalize the relevant semantic context and instantly receive a matching color palette. Our method can handle a single or multiple words, as well as any type of words, such as objects, locations or emotions.

We start with a large vocabulary of 100,000 frequently used words that we determine from Google n-grams, a byproduct of Google's book scanning project [1]. We then download for each word 60 related images using Google Image Search and store them in a database.

The images and related words are used to learn associatins between words and colors using a large scale statistical method [2, 3]. Given a specific word, the method employs a statistical significance test to assess whether its related images contain significantly more (or less) pixels of a given color. The link between a word and a color is expressed in the form of a normalized z value. We precompute these z values for all words and all colors in a color histogram.

The color palette extraction takes as input one or multiple words and looks up the respective significance distributions from a database. The palette's colors are then chosen so that they best represent the dominant regions of the words' significance distributions. We can add optimization constraints such as a hue template to assure a palette's color harmony.

We also demonstrate that our framework is able to extract color palettes from entire texts such as Wikipedia articles. This frees artists from extracting relevant keywords if they have, e.g., a project description in the form of a paragraph. Instead, they can just copy and paste the entire text and automatically obtain a matching color palette.

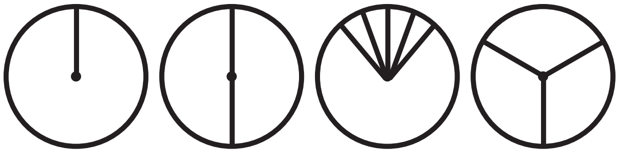

Figure 1. The four hue templates used for this research project. We do not use more complicated templates as they have been found to be less preferred [9].

This article is structured as follows. We begin with a state-of-the-art, outlining a brief history of color palettes and describing existing solutions. We then explain how we construct a large vocabulary of 100,000 words and how we acquire 6 million related images. The next section outlines the statistical framework and shows an example significance distribution. We then first focus on palette extraction based on a single word and then extend to entire texts. Finally we discuss the results on the basis of a psychophysical experiment and show that our palettes are slightly more preferred by observers than the corresponding palettes from Adobe Kuler [4]. Additionally, we demonstrate that the framework can estimate a palette's rank, which is important if multiple palettes for a word have to be presented to the user in a meaningful order.

Related Work

Early systematic studies on how colors relate to each other date back to Newton and Goethe in the 18th and early 19th century [5]. Goethe proposed a color wheel where colors are arranged in a way so that opposite colors "reciprocally evoke each other in the eye". Today, there are many different definitions of color wheels, such as Hering's color wheel based on opponent colors (red-green, blue-yellow) [6], Itten's color wheel based on red-yellow-blue primaries [7], or the HSV color wheel based on red-green-blue primaries.

Color wheels can be used to define a group of colors that harmonize well with each other. Itten proposes several hue templates such as *Dyads* (diametrically opposed colors in the color circle) or *Triads* (three colors that form a equilateral triangle on the color circle) [7]. These two templates correspond to the second and forth template reproduced in Figure 1, respectively. Templates can be rotated on the color wheel as long as the relative positions between the colors of the template remain constant. Hue templates for harmonic color themes were also studied by Matsuda who deduced similar templates from fashion questionnaires given to women [8].

Harmonizing colors defined by a hue template depend on the underlying color wheel. Let us exemplarily consider the color red; the diametrically opposed color is green on Hering's opponent color wheel, but cyan on the HSV color wheel based on RGB primaries. Thus, some color palette tools let users choose from different color wheel options [10]. In this work we use the HSV color space based on RGB primaries.

Color harmony is a controversially discussed topic and existing theories are continuously challenged and refined with psychophysical experiments [**?**]. O'Donovan et al. [9] approached the topic differently and analyzed 100,000 color palettes and their community ratings from Adobe Kuler [4]. The study concludes that only very basic hue templates with a small range of colors, typically 2 to 3, are preferred. This is the reason why we focus our experiments on basic hue templates and discard complex templates with 4 or more basic hues. Figure 1 illustrates the four hue templates used in this article.

There are numerous tools and services available to create color palettes [4, 11, 12, 13, 10]. They offer two different strategies to create palettes: the manual creation of a palette from scratch or the automatic extraction of a palette from a given image. Adobe Kuler [4] is a quite popular website where users can use both ways to create color palettes. The Kuler community contributed more than 800,000 palettes that have tags, comments and ratings. Color Hunter [11] is the second webpage we want to highlight due to a special feature. Users can enter a keyword, which is used to query images from Flickr. Then, one palette is extracted from each queried image and returned to the user. Instead of extracting a palette from a single image, our method uses a data-mining approach to extract a palette from many images related to a keyword. This makes our approach much more robust against wrong annotations and allows to extend the palette extraction to entire texts.

There is no rule to determine the number of colors of a palette, because it is the artist's choice how many colors are necessary for a given project. However, many tools such as Adobe Kuler [4] default on 5 colors. Even though our method is potentially able to handle any number of colors, we follow the community's trend of 5-color palettes, because it makes our approach more comparable to existing solutions.

Lindner et al. proposed a statistical method to link semantics with colors and used it for automatic color naming [2, 3]. The particularity of this approach is its small computational footprint, which makes it suitable for large-scale applications like this project. We thus use the publicly available code[1].

Building a Database

Our goal is to extract a color palette from a single word or an entire text. In order to build a fast system, we want to precompute statistics (see following Section) as much as possible. As we do not know in advance which words the user might input into the system we need to prepare a very large vocabulary of as many frequently used words as possible.

Google n-grams is a service from Google's book scanning project and its raw data is published under a Creative Commons License [1]. It contains the yearly count of n-grams (a sequence of n words) as found in all digitized books. For our application we focus on single words, i.e., 1-grams, and discard words that contain other characters but letters. Further, we consider only words from English books of the last 20 years. We then count the occurrences of all words and keep the 100,000 most frequently used words as our vocabulary. The last three words in this list are *Bayswater*[2], *turbidite*[3], and *trabalho*[4], meaning that anything more common is part of our vocabulary.

We use Google Image Search to download 60 related images for every word in the database, totalling 6 million images. We assume that the images are encoded in sRGB color space unless differently indicated.

Statistical Image-Mining

The goal in image-mining is to extract useful information from a large database of images with metadata. In our context the metadata are the words that were used to query the images. The information we want to extract is how each word relates to color. We therefore employ the framework of Lindner et al. [2, 3] for automatic color naming.

The framework computes for a given word its significance for an image characteristic as a standardized z score. It uses the Mann-Whitney-Wilcoxon test that determines whether two sets of random variables have equal medians or not [14]. In our case, the first set contains characteristics of all images queried with a given word and the second set contains the characteristics of all other images. For interested readers a mathematical explanation of the statistical framework is provided in the Appendix or Lindner et al. [3].

The characteristic employed in this research project is a color histogram in HSV color space with 16 equidistant bins along each dimension. Figure 2 shows the significance distribution for the word *grass* in a slice orthogonal to the v axis in HSV color space ($v = 0.5$). The image on the left hand side indicates each bin's color for a better orientation in the hs-plane and the right hand side shows the corresponding significance values. It is visible that greenish colors are highly significant for the word *grass*. The complete significance distribution is a volume of 16 such slices for all quantized v values.

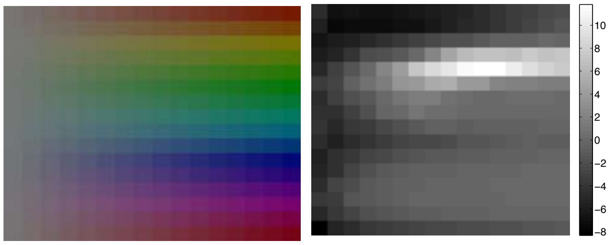

Figure 2. *An example slice ($v = 0.5$) of the significance distribution for the word $grass$. Left: the histogram bin's colors in the hs-plane. Right: the corresponding significance values. Greenish colors are dominantly significant for the word $grass$.*

Palette Extraction from a Single Word

We choose to extract palettes with 5 colors, because we are comparing our results against Adobe Kuler where 5-color palettes

[1] `http://ivrg.epfl.ch/research/ColorThesaurus.html`

[2] An area of central London.

[3] A geological formation from underwater avalanches.

[4] The Portuguese translation of *work*.

are the default. In order to create a color palette, we have to select 5 colors that are a good representation of a word and its underlying distribution in HSV space. We thus need a score that is high if the palette's colors are placed in areas of high significance.

Let $\mathbf{c} = [h s v]'$ be a color triplet in HSV space and $\mathbf{c_n}, n \in \{1 \dots 5\}$ the 5 colors of a palette. We then define the score s of a palette given a distribution $z(\mathbf{c})$ as:

$$s(\mathbf{c_n}) = \sum_{\mathbf{c}\in \mathrm{HSV}} z(\mathbf{c}) \cdot \max_{n=1\dots5} \left[\exp\{(\mathbf{c}-\mathbf{c_n})^2/\sigma^2\}\right] \qquad (1)$$

The max operator assures that it is not optimal to place all palette colors at the same location of maximum significance and enforces a certain spread. The sum operator integrates over the complete color space. We experimentally set $\sigma = 0.1$, but did not experience it to be a very sensitive parameter.

The score has to be optimized under the constraint that the palette's colors fulfill a given hue template. The optimal colors C^P and their score S^P for a given palette $P \in \{$⏀, ⌽, ⊛, ☮$\}$ are:

$$C^P = \operatorname*{argmax}_{\mathbf{c_n}\in P} s(\mathbf{c_n}) \qquad S^P = \max_{\mathbf{c_n}\in P} s(\mathbf{c_n}) \qquad (2)$$

Just as in Adobe Kuler, we add offsets to the color's s and v values in order to create some diversity. The hue templates are defined according to Table 1. All templates have the same 3 degrees of freedom, which are h, s, and v, respectively. The ⊛ template has an additional parameter α, which is the angle between the 5 equidistantly spread hue angles.

⏀			⌽		
h	$s-0.3$	$v+0.05$	h	$s-0.1$	$v+0.3$
h	s	$v+0.3$	h	$s+0.1$	$v-0.2$
h	s	v	h	s	v
h	$s-0.3$	$v+0.3$	$h+0.5$	$s+0.2$	$v-0.2$
h	s	$v-0.2$	$h+0.5$	s	v
⊛			☮		
$h-2\alpha$	$s+0.05$	$v+0.1$	$h+0.33$	$s-0.1$	v
$h-\alpha$	$s+0.05$	$v+0.05$	h	$s+0.1$	$v-0.3$
h	s	v	h	s	v
$h+\alpha$	$s+0.05$	$v+0.05$	$h+0.66$	$s+0.1$	$v-0.2$
$h+2\alpha$	$s+0.05$	$v+0.1$	$h+0.66$	s	$v+0.3$

Table 1. Definitions of the four hue templates in HSV color space. Similar to Adobe Kuler the s and v values have offsets to increase diversity. Hue angles wrap around the boundaries of the interval [0 1].

We search the entire parameter space in order to find the globally best palette. In the current implementation we limit the optimization to 8 quantized values per parameter to reduce the computation time to approximately 0.5 seconds per palette. We did not observe any visible quantization artefacts during our experiments, but if necessary the optimization can be refined at the expense of computation time or a more elaborate optimization strategy can be implemented.

Figure 3 shows example palettes for the word *cherry*. The score S^P and the palette template P are indicated to the left of the palette. The highest score is achieved for the ⏀ template with different shades of cherry red. The following templates add leave green and sky blue colors, respectively. The lowest score is achieved for the ⊛ template, because *cherry* does have a narrow peak along the hue axis, which penalizes a spread along this dimension.

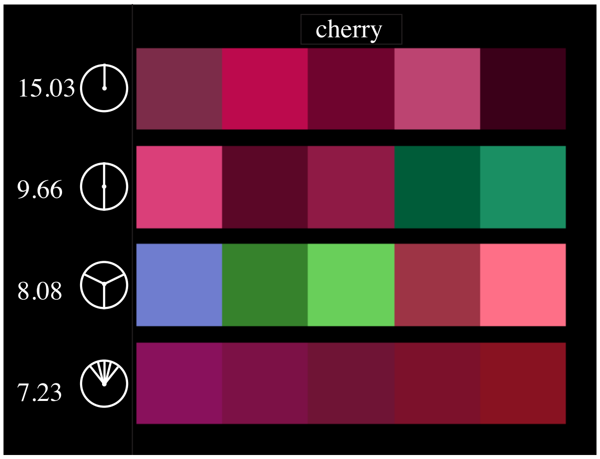

Figure 3. *All extracted palettes for cherry. The score S^P and the hue template P are indictaed to the left of each palette.*

If no specific template is desired by the user, all options can be created and the one with the absolut highest score can be returned. Figure 4 shows some example palettes for the words *grass* ⊛, *plum* ⌽, and *love* ⏀, respectively.

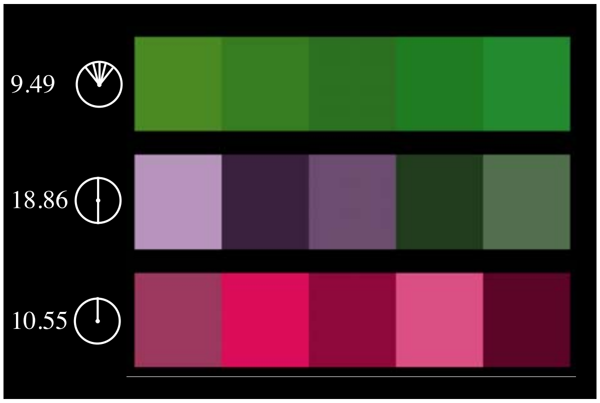

Figure 4. *Example palettes extracted with our automatic framework. The input words are from top to bottom: grass, plum, and love, respectively. The score S^P and the hue template P are indictaed to the left of each palette.*

Palette Extraction from Text

Instead of a single word we can also extract palettes from an entire text. In this case we compute a weighted average of the significance distributions of all words in the text, where the weight for each word corresponds to its relative frequency. If a text has a dominant topic that is represented with enough words, the final distribution is dominated by the respective words' distributions. All other words add up to a constant noise floor.

To demonstrate this use case we extract color palettes for the two introduction sections of the Wikipedia articles for *winter* and *swamp*. In both cases the optimal palette is of type ○. The results are reproduced in Figure 5.

Figure 6 shows a slice ($v = 0.5$) of the average significance distributions for both Wikipedia articles. The peaks are less dominant than in Figure 2 due to other words that contribute to the distribution, but there is still a dominant region for the articles on *winter* (middle) and *swamp* (right), respectivley.

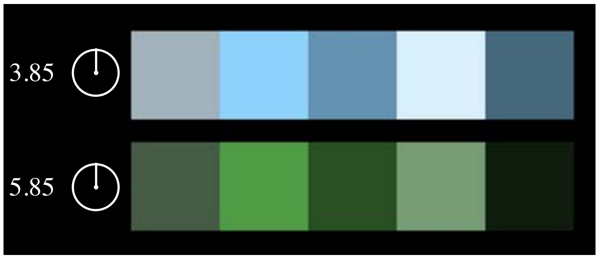

Figure 5. *Color palettes extracted from an entire text: the introduction sections of the Wikipedia articles of* winter *(top) and* swamp *(bottom). The score S^P and the hue template P are indictaed to the left of each palette.*

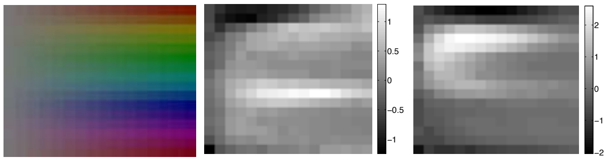

Figure 6. *Average significance distributions for the Wikipedia introductions of* winter *(middle) and* swamp *(right), respectively. The peaks are less pronounced than in Fig. 2, but still present.*

Psychophysical Evaluation

We evaluate the quality of the extracted palettes with an online psychophysical experiment. We compare our framework against Adobe Kuler, because it is a popular platform with a large database of palettes and annotated keywords.

We selected 30 words that approximately cover the whole color space: *bahamas*, *beach*, *carrot*, *cherry*, *cold*, *desert*, *dirt*, *dolphin*, *grass*, *green*, *jungle*, *lime*, *lion*, *love*, *nature*, *night*, *orange*, *plum*, *purple*, *raspberry*, *ski*, *sky*, *snow*, *sunset*, *tan*, *tree*, *warm*, *wildlife*, *wine*, and *winter*.

Using our automatic framework we compute color palettes for all words and hue templates and keep the three palettes with highest scores S^P. To make an equal comparison we need Kuler palettes for the same word and template combinations. Using Adobe Kuler's API we query and download up to 1000 palettes per word. Next, we find for each word the three palettes that best match the three previously determined templates using the distance of O'Donovan et al. [9]. The distance funtion adds up the hue angle distances between the hue angles of the present palette and the template. All palettes of the psychophysical experiment can be examined on our research webpage[5].

In the psychophysical experiment the observers saw the word on top and underneath the six color palettes (3 from our framework and 3 from Adobe Kuler, respectively) as shown in Figure 7. The two screenshots show the initial screen on the left and the screen after three selections on the right, respectively. The order of the palettes was randomized for every single observer. The observers were asked to rank the palettes according to how well they match the given topic. All observers saw the 30 topics in a different random order. In total, 25 observers participated in our study.

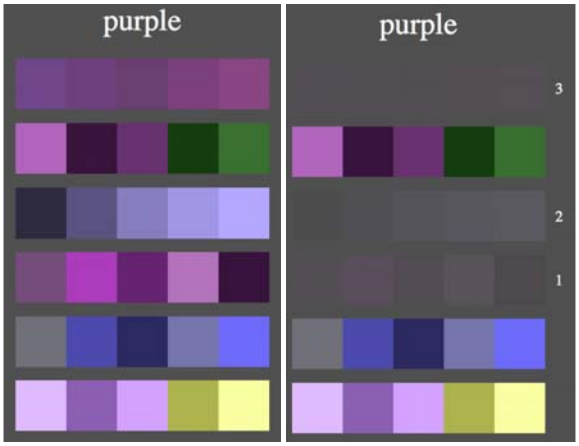

Figure 7. *Screenshots of the psychophysical experiment: initial screen (left) after the first three selections (right). The observers saw the word on top and six color palettes (three from our framework and Adobe Kuler, respectively) in random order below. The user were asked to rank the proposed palettes from 1 to 6 according to how well they match the given topic.*

Figure 9 shows the average rank for both methods and all words. Our proposed palettes have better average ranks than the ones from Kuler, but the difference is not statistically significant due to the relatively wide error bars. We believe that the large variation is in a large part due to the fact that ranking color palettes is highly influenced by personal taste. Still, this result is rather in favor of our approach, because our method is fully automatic and does not need user interaction; the color palettes on Adobe Kuler are manually created and annotated by the community.

We also evaluate the order that is predicted by our automatic framework. For this evaluation we consider only the palettes that stem from our framework. The score S^P (see Eq. 2) reflects how well an extracted palette represents the word, i.e. its underlying significance distribution. Our hypothesis is that a high score indicates a more preferred palette.

We compare the predicted ratings with the actual ratings of the observers using Spearman's rank correlation coefficient [15]. The coefficient is $\rho = 1$ if two sequences of ranks are perfectly correlated and $\rho = -1$ if they are inversely correlated. The average correlation in our psychophysical experiment is $\bar{\rho} = 0.49$ with a variance of $\sigma_\rho = 0.35$. The positive mean and comparatively smaller variance confirm that the score S^P is a meaningful estimation for a palette's preference rating.

A different and more depictive way to look at the rank order is the number of vicinal swaps necessary to transform the observer's ranking into the predicted ranking. For example, there are 2 vicinal swaps necessary to go from (312) to (123). Figure 8 shows a histogram of the number of vicinal swaps for all observers' rankings. In over 40% of the cases the estimated ranking identically matched the observer's ranking.

[5] `http://ivrg.epfl.ch/research/ColorPalettes.html`

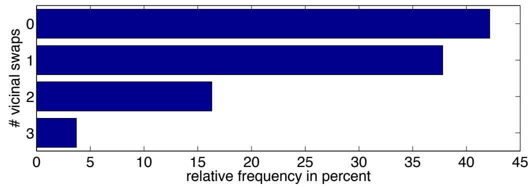

__Figure 8.__ Number of vicinal swaps necessary to go from the observer's ranking to the predicted ranking. In over 40% of the cases the predicted and the observer's ranking are identical.

Conclusions

We present a novel automatic method to create a color palette based on words. This is different from previous approaches where users have to manually create palettes or extract them from a single image. We believe that our text based interface enables artists and designers to find palettes more easily, because they can freely express their ideas and the emotions they wish to evoke in the minds of observers.

Our framework is based on a highly scalable statistical framework that we use to learn associations between colors and words. We pre-compute these association for a vocabulary of 100,000 words from a database of 6 million tagged images. The associations are represented in the form of a distribution of significance values in color space.

The significance distribution of a given word is then used to determine the most representative colors for a palette. The user can optionally choose a specific hue template to achieve a desired color harmony.

In the future work we want to investigate methods from natural language processing such as latent semantic analysis to further improve the palette extraction from text. Further it is necessary to discuss with artists and designers in order to refine our framework and realize a solution that is even more adapted to their needs in daily work life. Also it is interesting to compute statistics for other languages but English. This is easily feasable, because our framework is fully automatic and does not require the operator to speak the languages that are processed.

Acknowledgments

We thank Océ-Print Logic Technologies, Paris (Canon Group) for financing the author's PhD thesis that laid the foundation for this ongoing research and especially Christophe Leynadier and Dr. Maria-Valezzka Ortiz-Segovia for their support for the psychophysical evaluation. We also thank Dr. Nicolas Bonnier for all his support throughout the first 3 years of the author's PhD thesis.

Appendix

Mathematical explanation of the statistical framework

Let $w \in \mathbb{W}$ a word of the vocabulary $\mathbb{W}$, $\mathbb{I}_w$ the set of its associated images and $\mathbb{I}_{\overline{w}} = \{I : I \in \mathbb{I}_{w_i}, w_i \in \mathbb{W} \wedge w_i \neq w\}$ the united set of all images I not associated with w. We then compute for all images a specific characteristic, e.g., the bin count for a single histogram bin, resulting in the disjoint sets $\mathbb{C}_w$ and $\mathbb{C}_{\overline{w}}$ containing all characteristics of images annotated with w and all other images, respectively.

For the Mann-Whitney-Wilcoxon test [14, 16] all values in the combined set of $\mathbb{C}_w$ and $\mathbb{C}_{\overline{w}}$ are sorted in increasing order. Then the positional indexes of the values from the first set are summed up, yielding the ranksum statistic T.

For an illustrative example we assume $\mathbb{C}_w = \{1.2, 0.8, 1.5\}$ and $\mathbb{C}_{\overline{w}} = \{0.3, 1.1, 0.5, 0.9, 0.1\}$. The sorted combined set is $\overset{1}{0.1}, \overset{2}{0.3}, \overset{3}{0.5}, \overset{4}{0.8}, \overset{4}{0.9}, \overset{6}{1.1}, \overset{7}{1.2}, \overset{8}{1.5}$ with the positionional indexes stacked on top of the values. The ranksum of the first set's values is $T = 4 + 7 + 8 = 19$.

Usually the null hypothesis for this test is that the values in the two sets are equally large. In this case the expected mean and variance of the ranksum T are:

$$\mu_T = \frac{N_w(N_w + N_{\overline{w}} + 1)}{2} \qquad \sigma_T^2 = \frac{N_w N_{\overline{w}}(N_1 + N_{\overline{w}} + 1)}{12} \tag{3}$$

where N_w and $N_{\overline{w}}$ are the cardinalities of the sets $\mathbb{C}_w$ and $\mathbb{C}_{\overline{w}}$, respectively.

The mean and variance can be used to normalize the ranksum statistic and obtain the z score that is used throughout this article:

$$z = \frac{T - \mu_T}{\sigma_T} \tag{4}$$

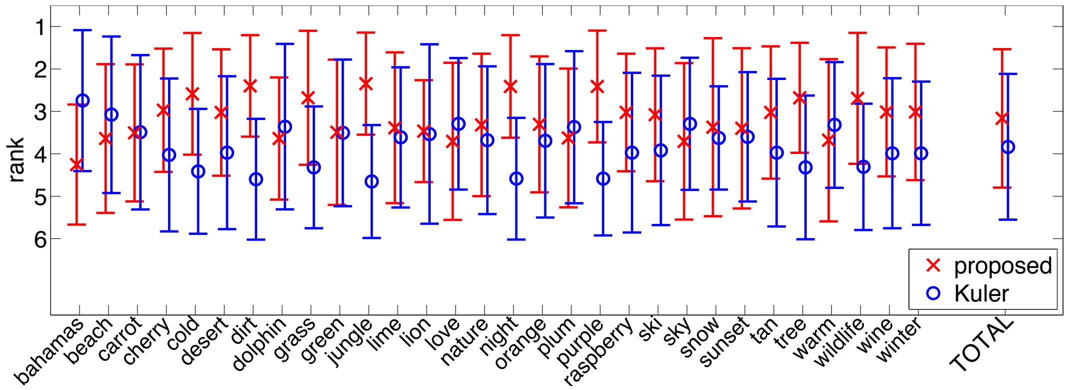

__Figure 9.__ Result of the psychophysical experiment with 25 observers: Mean ranks for all words and both methods. Our proposed approach performs better on average, but the difference is not significant. This is still a promising result considering the fact that our method is fully automatic whereas Adobe Kuler relies on manually created palettes from the web community.

The normalized ranksum for the previous example is:

$$z = \frac{19 - 13.5}{3.35} \approx 1.64 \tag{5}$$

and the positive z value indicates that the first set's values are significantly larger than the ones form the second set.

Our HSV histograms contain 16^3 bins and we consider each bin as a characteristic on its own and perform the statistical significance test for each of them, resulting in a distribution as shown in Figure 2.

References

[1] Google n-grams. http://books.google.com/ngrams/, last checked Apr. 2013.

[2] A. Lindner, B. Z. Li, N. Bonnier, and S. Süsstrunk, "A large-scale multi-lingual color thesaurus," in *IS&T CIC*, 2012.

[3] A. Lindner, N. Bonnier, and S. Süsstrunk, "What is the color of chocolate? - extracting color values of semantic expressions," in *IS&T CGIV*, 2012.

[4] "Adobe Kuler." https://kuler.adobe.com, last checked Apr. 2013.

[5] J. W. von Goethe, *Theory of Colors*. 1810.

[6] E. Hering, *Zur Lehre vom Lichtsinne*. G. A. Agoston, 1878.

[7] J. Itten, *The Elements of Color*. John Wiley and Sons, 1970.

[8] Y. Matsuda, *Color Design*. 1995.

[9] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Color compatibility from large datasets," in *SIGGRAPH*, 2011.

[10] B. J. Meier, A. M. Spalter, and D. B. Karelitz, "Interactive color palette tools," *IEEE Comput Graph Applications*, vol. 24, no. 3, pp. 64–72, 2004.

[11] "Color Hunter." http://www.colorhunter.com, last checked Apr. 2013.

[12] "CSS Drive." http://www.cssdrive.com/imagepalette/, last checked Apr. 2013.

[13] "Pictaculous." http://www.pictaculous.com, last checked Apr. 2013.

[14] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[15] C. Spearman, "'Footrule' for measuring correlation," *British Journal of Psychology*, vol. 2, no. 1, pp. 89–108, 1906.

[16] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947.

Author Biography

Dr. Albrecht Lindner is a postdoctoral researcher at ADSC in Singapore and works on large scale statistical image processing. He completed his PhD in spring 2013 under the supervision of Prof. Sabine Süsstrunk at EPFL, Switzerland. He holds two Master titles in electrical engineering and signal processing from the University of Stuttgart (Germany) and Telecom ParisTech (France), respectively.