IMAGE LEVEL COLOR CLASSIFICATION FOR COLORBLIND ASSISTANCE

Thomas L. Fuller and Amir Sadovnik

Lafayette College
Department of Computer Science
Easton, PA, USA

ABSTRACT

The advancement and proliferation of augmented reality lends itself to the development of novel techniques for assistive technologies, especially in the realm of computer vision. By enhancing a certain part of the view of a person with visual impairment we can assist them in different tasks. In this work we develop an algorithm to assist people who suffer from color blindness. We first examine different methods for pixel level color classification to select the one that works the best. We then improve the color classification rate by optimizing the labeling over the whole image using graph cuts. Finally, we develop an implementation of the algorithm which can run in real time on Google Glass and show how it can assist those suffering from color blindness.

***Index Terms*—** color classification, graph cut, assistive computer vision

1. INTRODUCTION

Assistive technologies aim to improve the quality of life for disabled, elderly, and even healthy people who struggle with everyday actions. As computer algorithms advance it is possible to use them to develop better assistive devices. While there are many areas that fall under the umbrella of assistive technologies, one area of interest is the area of assistive computer vision. Assistive computer vision looks to provide assistance for those who have difficulty seeing on their own.

In this work we aim to assist people who suffer from color vision deficiency (CVD) (see Fig. 1). CVD is the inability or decreased ability to see or perceive certain color differences under normal lighting conditions [1]. The two major classifications of colorblindness are red-green color blindness and blue-yellow color blindness. Although each of these classifications refers to the colors which are not distinguishable, they cover multiple types of color blindness with their own unique perceptions of other colors as well. In very rare cases, monochromacy can occur, which is the perception of a single color or no colors with perception based only on variations in brightness. While color blindness is usually inherited, other causes can be brain damage or retinal damage caused by overexposure to ultraviolet light. It is most commonly caused by a mutation on the X chromosome, and therefore affects an estimated 8% of the male population and only 0.5% of the female population.

There is no current cure for color blindness. Therefore, other methods for alleviating the difficulties which arise from CVD need to be found. The method should allow a color blind person to be able to differentiate between the colors they usually cannot. Ideally this should be done while the user is interacting with his environment, in real time, and will not require the user to look away from the scene. Augmented reality technologies such as the Google Glass, lend themselves for exactly this type of application. That is, by classifying each pixel to its correct color, the algorithm can highlight certain parts of a scene a user is currently looking at without them needing to look away.

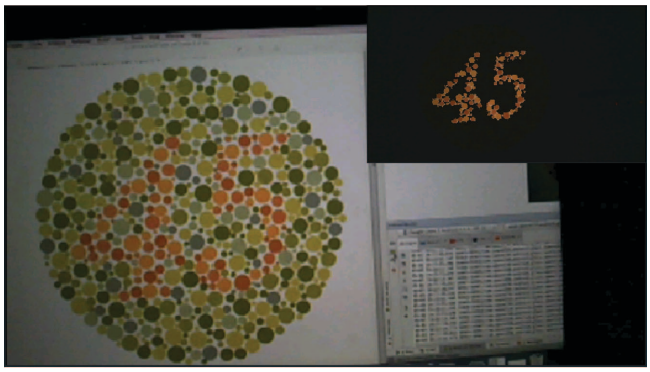

Fig. 1. In this work we develop a color classification algorithm which can run on Google Glass in real time and assist people with CVD. The image shows the original image (large) and the image shown to the user on the Google Glass (highlighting the color red) is shown on the top right. The advantage of our color classification algorithm is that instead of classifying each pixel individually we optimize the entire image classification using graph cuts. We show that using graph cuts significantly increases the color classification rate.

Automatic color classification is not a trivial problem. Due to the nature of colors as we perceive them, certain colors or shades may be falsely identified by an automatic classifier. This is partly because when a color is perceived by humans, it is dependent on the surrounding lighting on the object. For instance, a dark gray that is picked up by the camera may be identified as achromatic (grayscale), but in reality the color could be a very dark or unsaturated shade of blue. Additionally certain colors can be falsely identified for others when their saturation (chroma) is changed. (e.g. pink might be identified as a brighter and desaturated red).

In this work, we explore the development of a Google Glass application in the area of assistive computer vision for users affected by color blindness. We first attempt to develop a state of the art image color classification algorithm by extending the Parametric Fuzzy Sets algorithm [2] for color identification by using graph cuts (as shown in Fig. 2). We show that this improves the ability to identify colors within the context of an image. The end result is an improvement in color naming by an average of 21.97% on the image level, resulting in a less noisy classification. We then implement our full model on Google Glass, and show through a pilot test with people who suffer from CVD that our system allows them to accomplish

Fig. 2. An overview of our color classification method. Given an image we use Parametric Fuzzy Sets to classify each pixel to a color. We the use graph cuts to reduce the noise and achieve a more accurate color classifer.

tasks they previously were not able to.

2. PREVIOUS WORK

There have been a few works which have tried to either automatically classify colors by their names, or attempt to provide a useful tool to assist people with CVD. However, in this work we attempt to achieve both by first developing a state of the art method for color classification and then implementing it on a mobile wearable device as a tool for people with CVD.

2.1. Color Classification

Color classification has been addressed in a few previous works with two main approaches taken. First, some works attempt to use different machine learning algorithms to learn color names from images. For example, Van De Weijer et al. [3] provide a color identification approach that attempts to map the L*a*b* and RGB values to color names from Google Image search results. The colors were learned using probabilistic latent semantic analysis (PLSA), which allows for each L*a*b* value to be mapped to probabilities for each color name. The PLSA method creates word-topic distributions $p(w|z)$ for each pixel w within a set of images, where the color label is z. Pixels are then classified using a support vector machine.

Attempting to improve on the results of [3], Schauerte et al. [4] take a similar approach but use supervised latent Dirichlet allocation (SLDA). They manage to get slightly better results by using more sophisticated saliency detection and the advantages given by SLDA.

An alternative approach to color classification is to use a parametric model. For example, Benavente et al. [2] have developed a model for automatic color naming based on parametric fuzzy sets. With colors in CIE L*a*b* color space, the lightness value L, is first used to determine which of the 6 chromaticity planes will be used for the computation. Chromaticity planes contain different evaluation parameters for the model. Then, with the correct chromaticity plane selected, the a and b coordinates are used to determine the position on a triple sigmoid elliptical (TSE) function. The TSE function is a composition of 3 sigmoid functions and an elliptical function which determine the area in a, b space where the color lies. The method is both fast and space efficient.

SalahEldeen et al. [5] combine both methods (machine learning and parametric models). They attempt to learn the parameters of the TSE function directly from data. Although they do not receive a major difference in the results from the original TSE, they still manage to show a small improvement.

In this work our focus is not on the original pixel classification algorithm but more on examining if performing image level classification vs pixel level classification will be advantageous, and implementing the system to assist people with CVD. Therefore, we first wish to select a pixel level classification method which is both accurate and fast. In Sec. 3.1 we examine two pixel level classification methods (one learned and one parametric) and choose the one which performs best on both those measures. We then show in Sec. 3.2 how adding graph cuts improve the results.

2.2. Colorblind Assistance

Currently there exist a few approaches within the areas of image processing and computer vision which serve to alleviate the problems associated with CVD. These developments range from image transforms which can recolor an entire image, to specific hardware implementations, all with the end goal of providing users with the ability to distinguish between colors that would otherwise be excluded from their perception.

One way an image can be adapted for people with CVD is to shift the hues which a person cannot distinguish into a different range. This is done through a process called Daltonization. The Daltonization algorithm allows for scenes to express detail not normally perceived by a person with CVD by altering the image so that all colors fall within the user's perception gamut. This allows the user to differentiate between the colors in the image, providing the ability to comprehend scenes they might not be able to otherwise [6]. The Daltonization algorithm has also been improved upon using Chun-Rong Huang's method [7], which attempts to re-color the image by clustering colors into regions known as key colors. These clusters are then mapped into a CVD color space based on the particular type of CVD, and colors are shifted accordingly. This reduces the amount of colors that need to be processed during runtime.

The ColourPopper technique from Flatla et al. uses a user-selected highlighting of a popout color to increase the speed and accuracy for completion of Hardy Rand and Rittler (HRR) Plate tests [8]. The user first selects a popout color. Then, all pixels which do not match the popout color, have their respective luminance values reduced within the CIE L*u*v color space. This forces all colors besides the desired one to appear darker, causing the user-selected color to pop out. The technique has been compared against other color identification techniques for the purpose of helping users with CVD. They compare each technique by measuring the length and accuracy of completing a HRR Plate Test. In their results they found that ColourPopper led to the shortest mean completion time for each task, while still providing 98.59% accuracy.

Efforts have been made in the growing area of augmented reality. With wearable technology like Google Glass and Microsoft HoloLens, applications to assist those with CVD have already started being developed. Chroma, an application for the Google Glass device, has been developed by Tanuwidjaja et al. [9]. Chroma is a wearable augmented-reality system which provides the user a filtered image of the current scene. These filters apply different accessibility options like color highlighting and high contrast, outlining, and Daltonization modes.

In this work, we will be improving upon the color identification approach used in the Chroma application. Chroma [9] is focused

more on the user experience and the application itself and less on the color classification technique which uses a simple thresholding method. We attempt to to provide a more accurate color classification using graph cuts. In addition to this, we choose to use the ColourPopper technique for color highlighting, which improves the speed at which a user can distinguish a color. Using all of these methods, we develop an application for the Google Glass device that is an effective and more accurate tool to assist people with CVD.

3. METHOD

In order to develop a robust color classifier we first re examine two state of the art methods for color classification mentioned in Sec. 2.1. More specifically we examine one machine learning method using PLSA [3] and one parametric model using TSE [2] . By comparing each method's performance against a ground truth dataset, as well as considering the pros and cons of each approach, a method is chosen for the implementation on the Google Glass device.

3.1. Pixel Level Classification

In order to determine the efficacy of the color recognition methods, we tested both against The Fuzzy Color Naming dataset [2]. This dataset uses individual swatches for each color, and the colors are then named by participants in a study. The context of the color is disregarded, as the colors are presented in an isolated booth.

We divided the color swatches into three categories based on their ground truth confidence. We consider the confidence to be the percentage of people who agreed on a specific color name. We choose three confidence thresholds (0.3, 0.6, and 0.9) such that colors in the 0.9 and above are agreed to be a certain color by at least 90% of people while the other categories have less agreement.

When looking at the precision recall curves for all of the colors, it was determined that the two pixel level identification techniques performed very similarly. Performance on confident colors had nearly perfect precision recall curves (AUC of 1) for both methods. Additionally, for the uncertain (0.3) and fairly-certain(0.6) colors the performance was good and similar between the techniques. The similarity between the results suggests that both methods would be appropriate to use as pixel level identification techniques.

When considering which method to use, the TSE requires considerably less memory than the color learning lookup table to obtain results. The TSE also inherently accounts for all color values, due to the nature of fuzzy classification, whereas the lookup table requires a nearest neighbor to be calculated for the probability approximation.

We chose to use the TSE in our implementation for the aforementioned reasons. However, when running on natural images vs. color swatches many pixels were not labeled correctly. The mislabeling would vary in size and presence depending on the lighting conditions in the real world context. This is an area for improvement within the color identification problem that we sought to address. By improving the color classification, our application can provide CVD users with a better and more accurate user experience.

3.2. Image Level Classification

The pixel mislabeling problem occurs when certain pixels in an image are incorrectly classified due to missing the threshold probability constraint of the TSE. Because these pixels are not labeled correctly by the classification algorithm, the resulting image would only partially highlight pixels of a certain color. The holes within the pixel mask, can be seen in Figure 3 (a+b).

Fig. 3. The improvement gained when performing image level recognition using graph cuts. (a) Original image. (b) Pixels classified as blue before running the graph cut algorithm (Sec. 3.2). (c) Pixels classified as blue after using graph cuts. As can be seen the many of the holes that were not classified as blue in (b) because of lightning conditions were captured correctly in image (c) by using neighboring pixels.

Color	TSE	L*a*b* Dist	a*b* Dist	Improvement
Red	0.46	0.57	0.66	0.20
Orange	0.25	0.34	0.44	0.19
Brown	0.09	0.11	0.26	0.17
Yellow	0.33	0.47	0.65	0.32
Green	0.27	0.36	0.50	0.22
Blue	0.53	0.59	0.67	0.14
Purple	0.15	0.17	0.37	0.23
Pink	0.25	0.32	0.54	0.28

Table 1. Color classification results using the Jacard index (The higher the better). See Sec. 3.3 for details.

This labeling problem has traditionally been solved using energy minimization models. Within the context of an image with pixels P, let the neighborhood of pixels in the image be represented by $N \subset P \times P$. According to the theorem by Grieg et al. [10], an adjustment can be made in the pixel labelings by minimizing the standardized energy function:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in N} V_{p,q}(f_p, f_q) \quad (1)$$

In the energy equation, $D_p(f_p)$ is a function derived from the observed data, representing the cost of assigning a label to a pixel p. The function $V_{p,q}(f_p, f_q)$, represents the cost of assigning the labels f_p, f_q to the adjacent pixels, p, q, and is used to impose spatial smoothness.

The specific energy model that is useful within the context of color labeling is the Potts Interaction Energy Model. The Potts Model [11] can be described by the equation:

$$E(I) = \sum_{p \in P} ||I_p - I_p^o|| + \sum_{p,q \in N} K_{p,q} \times T(I_p \neq I_q) \quad (2)$$

where $I_p \in P$ are the unknown labels over the set of pixels P, and I_p^o are the labels observed with noise. The Potts interaction is specified by $K_{(p,q)}$ which is the penalty for label discontinuities between neighboring pixels. When $K_{(p,q)}$ is constant between all pairs of pixels, the term controls the smoothness of the classified pixels. The Potts model is solved by determining the max-flow of the system, which can be solved optimally in the case of binary labeling.

The data term, $D_p(f_p)$, is the cost of assigning a label to a pixel, and in Eq. 2 is represented as the difference between the known pixel labeling and the observed labeling. However, within our context, we are unsure of the true labeling. Since the value returned from the TSE function is a probability, it can be used to approximate the difference. Thus, the equation used for the data term, $D_p(f_p)$ is given by the following:

$$D_p(f_p) = -log(TSE(p)) \tag{3}$$

The pixel interaction term, $V_{(p,q)}$ which attempts to enforce smoothness is represented as $K_{(p,q)}$ when the labels disagree and zero otherwise. For the this penalty function we decide to use a value corresponding to the distance of the pixels' colors in the CIE L*a*b* color space. We test both the distance in (L, a, b) and (a, b), to determine which would be more effective. By using this distance as the interaction term, the more different a pixel is, the smaller the penalty for assigning two adjacent pixels different labels will be. We can then use the Graph cut algorithm to minimize the energy.

3.3. Improvement Using Graph Cuts

In order to show this improvement using graph cuts we wish to use a more realistic dataset than the color swatches. The dataset provided by van de Weijer et al. [17] uses images from eBay with manually labeled color masks. The dataset consists of 4 categories of objects; cars, shoes, dresses, and pottery. Each category has 12 images with mask labels for each of the 11 colors, totaling 528 images. The color recognition rate was measured by comparing the pixels labeled by the algorithms against the manually provided labels. We use the Jaccard index to quantify the results:

$$J = \frac{G \cap M}{G \cup M} \tag{4}$$

J is the Jaccard index, G is the set of pixels labeled by our graph, and M is the set of pixels labeled manually. The baseline for our comparison is found by processing the dataset with only the pixel level classification provided by the TSE. We then compare it to the results using the graph-cut algorithm with two different interaction terms, one which depends on the L*a*b* distance, and the other only on the a*b* distance ignoring the lighting condition. Performance is shown in Table 1, as well as a sample result in Figure 3.

Overall, there was a 21.97% improvement over the TSE in pixel classification using Graph Cuts where neighboring pixel edge weights were determined by their a*b* distance. This shows that lighting should be ignored for the pixel interaction. Two colors with the same a and b values and a different L value have the same color name more often than not. By looking at the colors in a*b* space, the interaction term is able to penalize giving different labels to pixels which are close, regardless of their L value.

4. GOOGLE GLASS IMPLEMENTATION

4.1. Application Design

The Google Glass application is designed to allow a user with CVD to select a color highlight. Using the TSE color scoring function described in Sec. 2.1 at the pixel level, the feed from the camera is processed. After every pixel in the image is scored, a graph is made using the Graph Cut algorithm described in Sec. 3.2. After the min-cut/max-flow algorithm has been performed on the graph, pixels matching the selected color are highlighted with the ColourPopper technique described in Sec. 2.2.

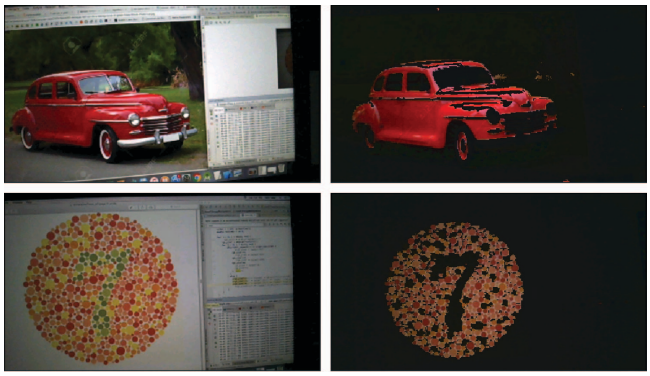

Fig. 4. Screenshots from our Google Glass Application. The left column present the original image the Google Glass user is observing, while the right column shows the images presented on the Google Glass screen classifying red.

4.2. Pilot Study Setup

We conducted a preliminary experiment to show how this application can assist people suffering from CVD. The participants in this study were volunteers from Lafayette College. Two groups of participants were created for the data collection. The control group was comprised of 4 students without CVD, whilst the test group consisted of 4 students who have CVD.

The participants in the study were given select plates from the 24 plate Ishihara test [12]. The 14 numerical test plates and 1 control plate were selected from the test. The response time for identifying each plate was measured from the time the plate was shown until the time a participant gave a response. Responses from individuals within the experimental group were recorded to determine the accuracy while completing the task. For the control group without CVD, the plates were presented one at a time without the aid of the Glass application. The average response time was measured over the 14 plates, and served as the baseline for time to complete the Ishihara test. The test group was then presented with the 14 Ishihara test plates in the same manner, but with the use of the Google Glass application. The response times were compared between members of the control and test groups.

4.3. Results

The results of the data collected from the pilot study are as follows. The average response time for non-CVD participants, who serve as the baseline was 1.39 seconds between the time the plate was shown and an answer was given. For the CVD participants who made up the test group, the average response time between plates was 3.6 seconds. Prior to the test of our application, the CVD participants were asked to identify the plates to see the accuracy without the assistance of our application. 1 was unable to identify 13/14, 2 were unable to identify 12/14, and 1 was unable to identify 11/14 of the Ishihara plates. However, the responses while using the application were 100% accurate across all users.

Although there is a small increase in response time between users using the application versus their non-CVD counterparts, this application still provided CVD users a method of achieving 100% accuracy on the Ishihara test. Using the ColourPopper method for highlighting, as well as Graph Cuts to remove holes in the image, proved to be a successful endeavor. Live results from the Glass device can be seen in Figure 4.

5. REFERENCES

[1] National Eye Institute, "Facts about color blindness," https://nei.nih.gov/health/color_blindness/facts_about.

[2] Robert Benavente, Maria Vanrell, and Ramon Baldrich, "Parametric fuzzy sets for automatic color naming," *Journal of the Optical Society of America*, 2008.

[3] Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, 2009.

[4] Boris Schauerte and Rainer Stiefelhagen, "Learning robust color name models from web images," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012.

[5] Hany M SalahEldeen, Robert Benavente, and Maria Vanrell, "A computational colour naming model trained on real-life images," 2009.

[6] Karl Rasche, Robert Geist, and James Westall, "Re-coloring images for gamuts of lower dimension," in *Computer Graphics Forum*, 2005.

[7] Chun-Rong Huang, Kuo-Chuan Chiu, and Chu-Song Chen, "Key color priority based image recoloring for dichromats," in *Pacific-Rim Conference on Multimedia*, 2010.

[8] David R Flatla, Alan R Andrade, Ross D Teviotdale, Dylan L Knowles, and Craig Stewart, "Colourid: Improving colour identification for people with impaired colour vision," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.

[9] Enrico Tanuwidjaja, Derek Huynh, Kirsten Koa, Calvin Nguyen, Churen Shao, Patrick Torbett, Colleen Emmenegger, and Nadir Weibel, "Chroma: a wearable augmented-reality solution for color blindness," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014.

[10] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 271–279, 1989.

[11] Yuri Boykov and Vladimir Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[12] Shinobu Ishihara, "series of plates designed as tests for colour-blindness," 1936.