Computers & Graphics 59 (2016) 143–150

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on SCCG 2016

Personalized 2D color maps

Nicholas Waldin [a,*], Matthias Bernhard [a], Peter Rautek [b], Ivan Viola [a]

[a] TU Wien, Austria
[b] KAUST, Saudi Arabia

ARTICLE INFO

Article history:
Received 14 February 2016
Received in revised form
17 May 2016
Accepted 10 June 2016
Available online 24 June 2016

Keywords:
Color
Perception
Color vision deficiency

ABSTRACT

2D color maps are often used to visually encode complex data characteristics such as heat or height. The comprehension of color maps in visualization is affected by the display (e.g., a monitor) and the perceptual abilities of the viewer. In this paper we present a novel method to measure a user's ability to distinguish colors of a two-dimensional color map on a given monitor. We show how to adapt the color map to the user and display to optimally compensate for the measured deficiencies. Furthermore, we improve user acceptance of the calibration procedure by transforming the calibration into a game. The user has to sort colors along a line in a 3D color space in a competitive fashion. The errors the user makes in sorting these lines are used to adapt the color map to his perceptual capabilities.

1. Introduction

Color maps are commonly used to convey data properties in height or heat maps. In a one-dimensional case the data is mapped onto a line in a color space, for example, RGB, CIELab, or a single color that changes in intensity. In two dimensions, either a plane is taken in a color space, e.g. RGB, or each axis corresponds to a color and a data value's color is a linear combination of the axes. Comprehension of two-dimensional color maps can be curtailed if the perceptual distance between data values mapped onto the color map is not consistent with the distance between original data values [1]. Fortunately, for people with normal color vision, despite having large differences in cone ratios, color perception is fairly similar [2], so standard perceptually uniform color spaces can be used. However, for users with a form of color deficiency, such as anomalous trichromats or dichromats, conventional color maps can raise several issues. First, there is the obvious problem of being unable to distinguish certain colors, such as red and green in the case of red–green blindness. The second issue is a change in perceptual distance. Because the perceptual distance between colors that are near each other can be very short, and for people with color vision deficiencies even look the same, the user will have difficulty in comprehending the data with these colors. As a result, he/she may overlook important features. This problem can be addressed by personalizing color maps. Our method extracts lines of color gradients from the color map and takes samples along it. These samples are displayed to the user as a sequence of squares, ordered randomly. The user must then sort these squares to achieve a smooth color gradient. An example of this task is shown in Fig. 1b, and an example of the sorting in Fig. 1a. Based on the errors the user makes, we personalize the color space by contracting and expanding areas of the color map where the user performs worse or better. Through this we can personalize any continuous color space.

2. Related work

Two-dimensional color maps are used in a large variety of tasks [4], and are also used with higher dimensional data with data being projected onto a two-dimensional color map. In this case, tasks fall generally into one of the two groups: either into identification and comparison of data points and clusters or lookup of classes and clusters. For people without color vision deficiencies often perceptually uniform color maps [1] are used. The choice of color map depends on the task, and there is research on how to choose the appropriate color map [5]. An area where color map alterations have been demonstrated as useful is the remapping of colors for people with color vision deficiencies, for example encoding colors as patterns [6] and completely altering the colors of the image, both for stylistic reasons [7,8], as well as scientific purposes [9,10], and in real-time [11]. A longer list of examples can be found in a recent survey [12]. There are also methods for calibrating monitors to people with color deficits in order to recolor images, such as [13,14], which will be touched in Section 7.3. However, the trend of personalizing visualization and color maps is gaining momentum. One approach uses human perception of faces to generate isoluminant color maps [15]. An image of a face is divided into two parts, one white and one black. The image is copied and the white and black areas reversed. The image and its copy are placed side by side and shown to the user, with the black areas set to a gray value, and the

---

[*] Corresponding author.
E-mail address: nwaldin@cg.tuwien.ac.at (N. Waldin).

http://dx.doi.org/10.1016/j.cag.2016.06.004
0097-8493/

Fig. 1. In (a) an illustration of a person sorting a line is shown. The top line is presented to the person. S/He moves individual squares in the line in an attempt to sort it. In the second line one square has been moved. In the last line all the squares have been correctly sorted. In (b) a screenshot of what is displayed in our method to the user in the center of the screen is shown. The black area between what is shown and the edge of the monitor has been cut away. The colors have been altered significantly for easier perception. (c) shows a screenshot of the game Blendoku [3] on medium difficulty. Menu, stage name, and similar have been removed. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

white areas to a color. Two images are used because, if the luminance is unequal, one of the faces will stand out more. The user alters the luminance of the colored area until he/she thinks the luminance of the colored area is the same as the luminance of the gray area. This is done repeatedly, each time with a different color. The settings are then used to create an isoluminant color bar. Another approach [16] shows a method whereby a one-dimensional color bar can be adapted to an individual's color perception by searching for small changes in the color bar. This method can adapt a color bar in 10 min using 15 sample points. At each sample point the color of the sample point is taken and copied to another position along the bar. This creates a dot along the bar, which the user must search for and find within 5 s. While this method can be used to adapt two-dimensional color maps, it does not scale well. If we expand this method to two dimensions, this would mean about 15^2 (225) points of interest. Furthermore, working in two dimensions increases the number of directions. Because the number of directions in two dimensions is infinite, it is imperative to select the least number of directions that can still give a useful result. In the case of an arbitrary plane, all axes of the color map and their combinations are of interest, because the change in color and perceptual distances along a combination may be very different from the changes along an axis of the plane. Therefore, both the x-axis, y-axis and the diagonal directions $[x, y] = [1, \pm 1]$ should be investigated, leading to a total of 4 directions. As a result, in this approach the amount of work is first squared (number of points) and then multiplied by 4, leading to an approximate testing time of 400 min, or 6 h 40 min. If we assume that their approach uses as many points as we have areas in our experiment (20), then their approach would use $10 \times (20/15) \times 4 = 53$ min. However, the user will require a break. If we assume that the user takes a 5 min break after every 10 min, then it will take a total of 70–75 min. Our work has a similar goal. However, we developed a method that scales better than two dimensions is less exhausting and more enjoyable. In contrast, our method can be completed in as little as 30 min, though some users take significantly longer. Furthermore, it is influenced by the game Blendoku [3], which has a huge player base, indicating that it is enjoyable. It is also not as exhausting. In the approach paper by Gresh [16], the user must constantly react within seconds. In our game the user progresses as fast as he/she wants.

3. Methodology

People dislike using software that adapts to the users because it takes time, effort, and is usually tedious and boring. Therefore we have drawn inspiration from the field of mobile gaming in order to overcome these issues. Our method is based on the game Blendoku, where the player must sort colors. In this game the player is given a figure consisting of squares. He/She must move squares with different colors onto the fields of the figure, such that the colors change from one hue to another between two different squares on the game figure. In our method, we do not include a puzzle aspect, as we only wanted to focus on the ability to distinguish colors. As such, we greatly reduced the difference in colors between tiles. A screenshot of our version can be seen in Fig. 1b, and Blendoku in Fig. 1c. In our game, the user is asked to sort a sequence of eight squares, the first and last cannot be moved. Using a pilot study with different color maps and methods of creating sequences, we analyzed what measurements could be used. We measured the number of incorrectly sorted squares, time spent comparing two colors, time needed to sort a sequence, and the number of times squares were moved. Only the number of incorrectly sorted squares was useful, as no other measurement correlated with it. Therefore, the method for adapting the color map uses only the number of errors made, and it attempts to alter the color map so that the areas of the color map with higher error are shrunk and areas with low error are expanded, increasing and decreasing the perceptual distance between data values. In this section, we will describe the probing and evaluation method. All RGB values are between 0 and 1 and in the format RGB=[R,G,B].

3.1. Forced choice statistics

Our method is essentially a yes-or-no forced choice method. In forced choice statistics, the user must choose between two options, in our case if a tile is to the right or left of another. The probability distribution of normal forced choice follows the curve in Fig. 2. When the user cannot determine which option to choose, he must guess. The threshold for noticing a difference is placed at 75%. See Mckee et al. [17] for details. In our case of a line, the function will not decline as quickly, because the user can use additional squares to compare. For example, he may not see the difference between squares 2 and 3 and squares 3 and 4, but may see a difference between 2 and 4. It needs to be pointed out that if a person can solve a line without making a mistake, for example because it is very easy, then our method cannot tell the degree of difficulty of perception in this area. Another aspect of this forced choice is the effect on an adaption of the color map based on measured user errors. If a color map is optimally adapted to a person, then the perceptual difficulty would be evenly distributed over the color map for this person, i.e., the probability of an error would be independent of the position on the color map. If we take the mean perceptual difficulty and alter all areas to have this difficulty, then we will have achieved an even distribution of perceptual difficulty. This is described in Section 3.5.