An Interactive App for Color Deficient Viewers

Cheryl Lau[1], Nicolas Perdu,[1], Carlos E. Rodríguez-Pardo[2],
Sabine Süsstrunk[1], and Gaurav Sharma[2]
[1]École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[2]ECE Dept. Univ. of Rochester, Rochester, NY, USA

ABSTRACT

Color deficient individuals have trouble seeing color contrasts that could be very apparent to individuals with normal color vision. For example, for some color deficient individuals, red and green apples do not have the striking contrast they have for those with normal color vision, or the abundance of red cherries in a tree is not immediately clear due to a lack of perceived contrast. We present a smartphone app that enables color deficient users to visualize such problematic color contrasts in order to help them with daily tasks. The user interacts with the app through the touchscreen. As the user traces a path around the touchscreen, the colors in the image change continuously via a transform that enhances contrasts that are weak or imperceptible for the user under native viewing conditions. Specifically, we propose a transform that shears the data along lines parallel to the dimension corresponding to the affected cone sensitivity of the user. The amount and direction of shear are controlled by the user's finger movement over the touchscreen allowing them to visualize these contrasts. Using the GPU, this simple transformation, consisting of a linear shear and translation, is performed efficiently on each pixel and in real-time with the changing position of the user's finger. The user can use the app to aid daily tasks such as distinguishing between red and green apples or picking out ripe bananas.

Keywords: dichromacy, color deficiency, smartphone app, shear transformation, color blindness.

1. INTRODUCTION

Seven to ten percent of males have some form of red-green color deficiency.[1] For these individuals, colors with reds and greens can look very similar even though these colors would appear very different to a person with normal color vision. Normal everyday tasks such as picking out clothes that match, choosing ripe fruit at the grocery store, or deciding when meat is cooked are more difficult for those with a color deficiency. Judging the color of LED indicator lights such as those on chargers or swipe access card readers can be frustrating. It could be hard to read a map or understand graphs and charts in presentations if the colors used are not distinguishable. These are common tasks that color deficient individuals face with some difficulty.[2,3]

We propose a smartphone app that helps color deficient individuals distinguish between problematic colors and judge the color of objects. Our app is based on the key idea of converting color contrasts in a static image to contrasts seen over time. As the user interacts with the device's touchscreen, our method continually changes the image colors. For example, for a color deficient viewer, the red and green apples in Figure 1 can be hard to distinguish. When viewing the apples in our app, the user can slide his finger around the touchscreen to make the red apples appear different from the green apples. As the user traces a path, the colors will change according to the sequence of shear transformations. For a particular path, the red apples will take on a sequence of colors that is different from the green apples. By learning the sequence of colors associated with the red apples, the user could eventually learn how to judge an apple's color.

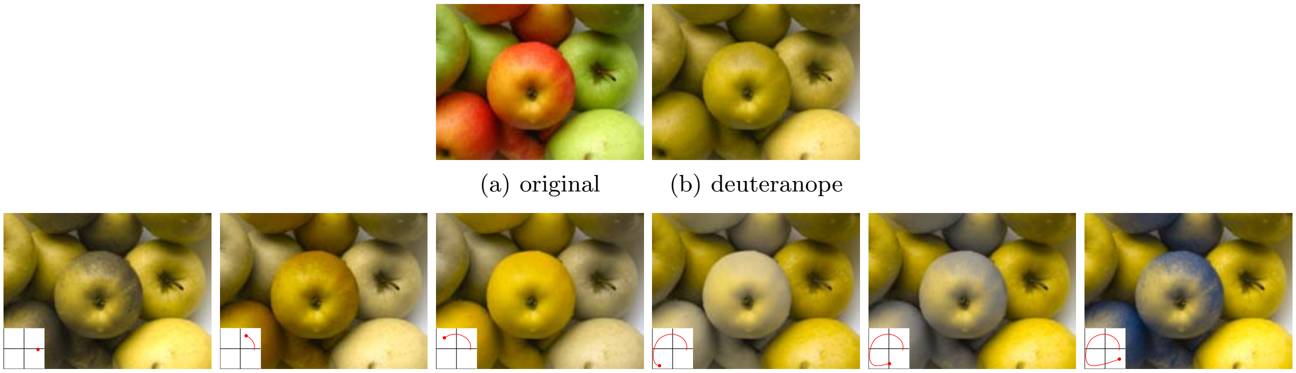
Although red-green color deficiencies are the most common form, there are several types of color deficiencies. People with normal color vision have three types of cones (L, M, and S), and color deficiencies are categorized by which cone is affected and to what degree. Anomalous trichomats have one cone whose absorption spectrum is shifted with respect to the spectrum of a normal viewer. In dichromats, one cone does not function correctly

Further author information:
Carlos Rodríguez-Pardo: E-mail: pardo@ece.rochester.edu, Telephone: 1 585 275 8122

Color Imaging XX: Displaying, Processing, Hardcopy, and Applications, edited by Reiner Eschbach, Gabriel G. Marcu,
Alessandro Rizzi, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 9395, 939512 · © 2015 SPIE-IS&T
CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2075162

because it lacks the light-sensitive pigment. Each of these categories is divided into three sub-categories based on which cone is affected. Protanomalous, deuteranomalous, and tritanomalous viewers are anomalous trichromats whose L, M, and S cones are affected, respectively. Likewise, protanopes, deuteranopes, and tritanopes are dichromats whose L, M, and S cones are affected, respectively. Those with a red-green color deficiency fall within the protan and deutan cases while the blue-yellow deficiency happens in the rare tritan case. Monochromats are those with two or three non-functioning cones, but this case is rare compared to red-green deficiencies. Our app is designed for the more prevalent anomalous trichromat and dichromat cases, enabling these viewers to visualize the third color dimension they lack.

Our method is based on a simple linear shear transformation that shears the data so that different information can be visualized with each shear. Consider a surface in LMS cone space containing the set of colors distinguishable by a dichromat. Brettel et al.[4] approximate such a surface as shown in Figure 2(a) for the protanopic case. For a protanope, or one whose L cone is affected, they simulate dichromatic vision by projecting colors along lines parallel to the L axis. This means that colors along the same line will project to the same color on the viewer's surface; these colors will look the same to the viewer. Our method applies a shear to lines of colors so that colors on the same line now project to different colors on the viewer's surface, enabling the viewer to differentiate between the colors. Different shears are applied as the user moves around the touchscreen, changing the color contrasts seen in the image. Our app allows color deficient viewers to interactively visualize color contrasts that normal viewers see as changes in color over time.



(a) original (b) deuteranope

Figure 1. **Interactive app for visualizing color contrasts.** The red and green apples in (a) can be hard to distinguish for an individual with a deuteranopic color deficiency (simulated in (b)). Our smartphone app allows users to interactively change the colors in the image to reveal potentially hidden color contrasts. In the second row, as the user moves a finger around the touchscreen (visualized in the bottom left corners), the red and green apples change colors, revealing the contrast between them. *Original image courtesy of Flickr user zaveqna.*[5]

2. RELATED WORK

Many recoloring algorithms have been developed to enable color deficient viewers to visualize contrast that is difficult for them to see. These methods recolor an image using optimization techniques or other algorithms to solve for new colors that exhibit better contrast for the viewer. Daltonize, online software developed by Dougherty and Wade,[6] stretches and projects red/green contrast onto the lightness and blue/yellow axes. Google offers a browser extension called Chrome Daltonize[7] which enables users to recolor webpages for their particular deficiency type. Wakita and Shimamura[8] present a method to recolor documents based on user specified contrast and naturalness constraints on document elements. With Jefferson and Harvey's interface,[9] the user can interactively recolor images with a slider. Our method similarly gives the user control over the recoloring and immediate visual feedback, but our method provides smooth color transitions in a continuous space rather than the discrete sampling implemented in their single slider. Machado and Oliveira[10] present a real-time and temporally coherent solution for recoloring images. Flatla and Gutwin[11] create models specific to individual users or situations (i.e. wearing colored glasses, being in sunlight) and recolor images based on those models. Sajadi et al.[12] overlay differentiating patterns to help viewers discriminate between color regions. Rasche et al.,[13] Kuhn et al.,[14] and Eynard et al.[15] present multidimensional scaling, mass spring, and Laplacian based optimization methods that solve for new image colors that preserve contrast and structure. For sharing documents, Hung and Hiramatsu[16] proposed the insertion of hatching patterns to allow different users to identify differences in color, while Rodríguez-Pardo and Sharma[17] proposed an adaptive methodology for visualizing graphics depicting qualitative data differences by efficiently generating a dynamic palette that is most distinctive for the specific observer deficiency and the number of distinct colors in the graphic. All of the recoloring methods, with the exception of Jefferson and Harvey's method,[9] produce a single optimized output image. In contrast, our app allows the user to *interactively generate different renderings to visualize contrasts as changes in contrast over time*. Also, many of the methods that involve optimization spend time computing a single best output, but with our simple, fast method, users can choose a rendering that works with a single swipe of their finger.

Smartphone apps exist to help color deficient users with daily tasks. Daniel Flück[18] posts a list of apps for naming colors, simulating color deficient vision, recoloring images, and finding colors that match or harmonize well together. The most readily available apps are those designed for mobile devices using the Android$^{\text{TM}}$and iOS$^{\text{TM}}$operating systems. DanKam,[19] the most publicized of these applications, operates by increasing the saturation and coarsely quantizing the hue with the goal of maintaining distinguishability between distinct hues. Color blindness correction[20] operates by enhancing color images by exaggerating the saturation and modifying the colors of regions likely to be confused with background regions. Enliven[21] and Color Unblinder,[22] on the other hand, alter colors and textures to exaggerate and make visible differences that a color deficient individual may miss. Similar to the published recoloring methods, these apps apply corrections to the image to achieve a single output image. By starting with the natural visualization, users of our app can start with what they see normally and then manipulate the colors smoothly through a real-time interaction.

3. METHOD

Our app transforms image colors according to the user's input. The user places his finger anywhere on the touchscreen to determine the origin of the 2D input coordinate frame. As the user drags his finger around the touchscreen, we update the image colors according the the user's position relative to the coordinate frame. For each user position (x, y), we apply a particular shear transformation so that as the user moves around the touchscreen, the image colors are continually and smoothly changing.

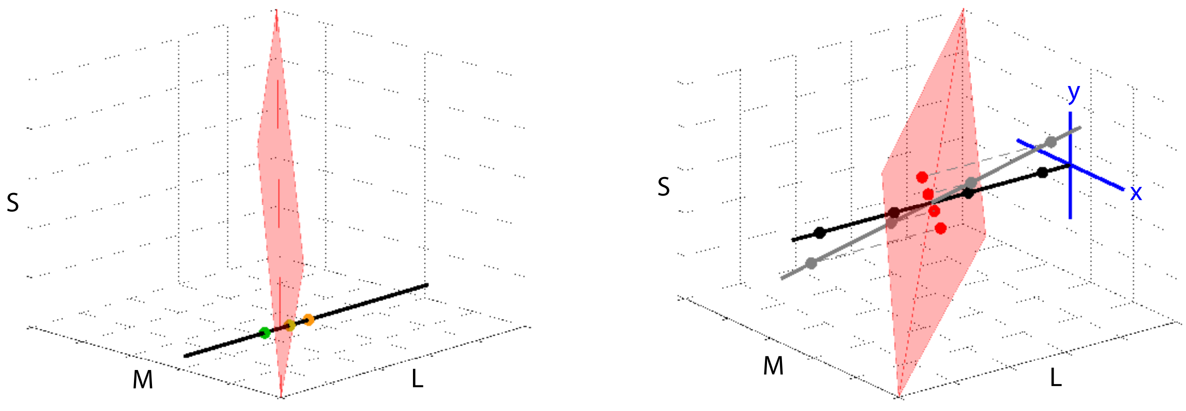
Our method is based on a simple linear transformation that shears the 3D data. When sheared, data points that previously projected to the same point now project to different points on a viewer's surface of distinguishable colors. Let us first consider the dichromatic case. Brettel et al.[4] approximate the surface of distinguishable colors for a dichromat as two planes hinged by the neutral axis. Figure 2(a) depicts this surface for a protanopic viewer and illustrates how protanopic vision is simulated. Brettel et al. simulate protanopic vision by projecting all colors along the L axis, the affected cone type for this viewer, to the planar surfaces. To allow color deficient viewers to visualize color contrasts along the affected axis, we shear the lines parallel to the affected axis so that the transformed points project to new colors on the viewer's surface as in Figure 2(b). In order to make sure colors on the viewer's surface do not change, we need to perform a translation after shearing. First, we convert the sRGB input image to LMS cone space. For each color $\mathbf{l}$ in LMS space, we transform it as follows to get the output color $\hat{\mathbf{l}}$:

$$\hat{\mathbf{l}} = \mathbf{S}\mathbf{l} - \mathbf{t}, \tag{1}$$

where $\mathbf{S}$ is the shear matrix and $\mathbf{t}$ is the translation vector. We then convert the output LMS color $\hat{\mathbf{l}}$ to sRGB space for display.

The shear matrix $\mathbf{S}$ is defined by the user position (x, y) so that lines parallel to the L axis are sheared for protanopes, lines parallel to the M axis are sheared for deuteranopes, and lines parallel to the S axis are sheared for tritanopes. For a protanope,

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ x & 1 & 0 \\ y & 0 & 1 \end{bmatrix}, \tag{2}$$



(a) protanopic simulation

(b) our method

Figure 2. **Our color transformation.** (a) Brettel et al.[4] describe a protanope's set of distinguishable colors as a surface (red) in LMS cone space. They simulate protanopic vision by projecting colors (green and orange) to the surface along lines parallel the the L axis, the axis of the affected cone. All colors along the line project to the same color on the surface and thus all look like that color. They define similar surfaces and projections for the deuteranopic and tritanopic cases. (b) We apply a shear and translation to constant color lines (black) so that the transformed colors (gray) project to different colors (red) on the viewer's surface, enabling the viewer to distinguish between the colors. The angle of the shear is determined by the user input specified by the coordinate frame (blue) and the line's intersection with the surface. Points on the surface remain the same.

for a deuteranope,

$$\mathbf{S} = \begin{bmatrix} 1 & x & 0 \\ 0 & 1 & 0 \\ 0 & y & 1 \end{bmatrix}, \tag{3}$$

and for a tritanope,

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

The translation vector $\mathbf{t}$ shifts the sheared lines so that points on the viewer's surface do not change. Specifically, this shifts the sheared line so that it intersects the viewer's surface at the same point the original line did. For a protanope,

$$\mathbf{t} = \begin{bmatrix} 0 \\ \mathbf{s}_l x \\ \mathbf{s}_l y \end{bmatrix}, \tag{5}$$

for a deuteranope,

$$\mathbf{t} = \begin{bmatrix} \mathbf{s}_m x \\ 0 \\ \mathbf{s}_m y \end{bmatrix}, \tag{6}$$

and for a tritanope,

$$\mathbf{t} = \begin{bmatrix} \mathbf{s}_s x \\ \mathbf{s}_s y \\ 0 \end{bmatrix}, \tag{7}$$

where $\mathbf{s} = (\mathbf{s}_l, \mathbf{s}_m, \mathbf{s}_s)$ is the simulated dichromatic color of the original color $\mathbf{l}$ calculated by the projection defined by Brettel et al.[4] We set the coordinate frame limits for the (x, y) user positions to $[-3, 3]$ for both axes in the protanopic and deuteranopic cases and to $[-\frac{1}{3}, \frac{1}{3}]$ in the tritanopic case.

We can easily apply the transformation in linear RGB space since the RGB to LMS conversion is described by a linear transformation. This makes the transformation simpler and faster. We add conversion matrices to Equation 1 to get

$$\hat{\mathbf{r}} = \mathbf{M}^{-1}\mathbf{S}\mathbf{M}\mathbf{r} - \mathbf{M}^{-1}\mathbf{t} \tag{8}$$

where $\mathbf{M}$ converts linear RGB colors to LMS space and $\mathbf{M}^{-1}$ converts LMS colors to linear RGB space. This allows us to apply the shear and translation directly to the linear RGB color $\mathbf{r}$ to get the output color $\hat{\mathbf{r}}$ in linear RGB space. At the beginning and end of the process described by Equation 8, we convert from and to sRGB space. This amounts to one matrix multiplication and one vector addition per pixel in addition to the sRGB gamma functions (forward and reverse), allowing us to efficiently implement this on the GPU for interaction at real-time rates.

Although we describe this transformation using an example of dichromatic simulation, this transformation also enables anomalous trichromats to visualize weak color contrasts that appear strong to color normal individuals. Protanomalous, deuteranomalous, and tritanomalous viewers, similar to their dichromatic counterparts, have limited distinguishability along the L, M, and S dimensions of color, respectively. The shearing operation will alter the values of the two unaffected color dimensions to create contrast between colors that only vary in the affected color dimension. With a simple shear and translation, defined differently for protans, deutans, and tritans, our app changes color contrasts for both anomalous trichromats and dichromats, enabling them to visualize contrasts apparent to viewers with normal color vision.

4. RESULTS

Our app allows color deficient users to visualize color contrasts that are immediately apparent to those with normal vision. The first two columns in Figures 3 and 4 show some example images and simulated versions for different viewer types. Deuteranopic and protanopic viewers do not sense the strong contrast between reds/oranges and greens that those with normal vision perceive as extremely different. They also have difficulty telling the difference between ripe yellow bananas and unripe green bananas. With our app, the color deficient user can interactively change the colors of the image to reveal such contrasts. Figure 3 shows results of a user changing the colors of images to reveal the strong red-green contrast. Over the course of the user's trajectory, the red pepper in the middle of the third image changes from a dark color to brownish-yellows to grays and finally to blue in the simulated images (using color names a person with normal color vision would use to describe the simulated images). This sequence of colors is similar for all the red colors. Likewise, the light green colors of the bananas and the tomatoes follow a sequence of colors that is unique for that shade of green. By associating a sequence with a user path, one could learn to judge the color of objects as well as discriminate between them. Figure 4 shows results of our method for different viewer types. Our app has settings for protan, deutan, and tritan viewer types. In the first two rows, we show output images simulated for a protanopic viewer in which the number six is revealed, and the foreground/background contrast is increased. In the last two rows, we show output images simulated for a tritanopic viewer in which the color difference between blues and greens is made more apparent, especially in columns 2 and 3. The coordinate frame is scaled by a factor of $\frac{1}{9}$ in the tritanopic case.

To validate the ability of the proposed methodology to help users to distinguish color contrasts we developed a game based evaluation. Games have been successfully used to evaluate the user performance in other scenarios.[31] The purpose of a game-based validation is twofold. A game motivate users to correctly complete a certain task, which otherwise may be trivial, repetitive, or boring. Moreover, a game is a convenient strategy to engage users with the capabilities of the method while exploiting the computational infrastructure created for the app.

The game designed for the validation is based on a matching experiment. In each trial of the game, the observer is presented with multiple color patches and asked to select a closely matching pair. To make the game more challenging and to better approximate the multiple colors in real scenes, we use eight patches for each trial of our matching game, selected as follows. Two points are first sampled on the dichromatic surface of distinguishable colors corresponding to the user (see Fig. 2(a)) with a spacing greater than 35 ΔE units in the perceptually uniform CIELUV color space. For each of the sampled colors, three colors each are then chosen along the axis of confusion for the corresponding dichromatic observer with a spacing greater than 15

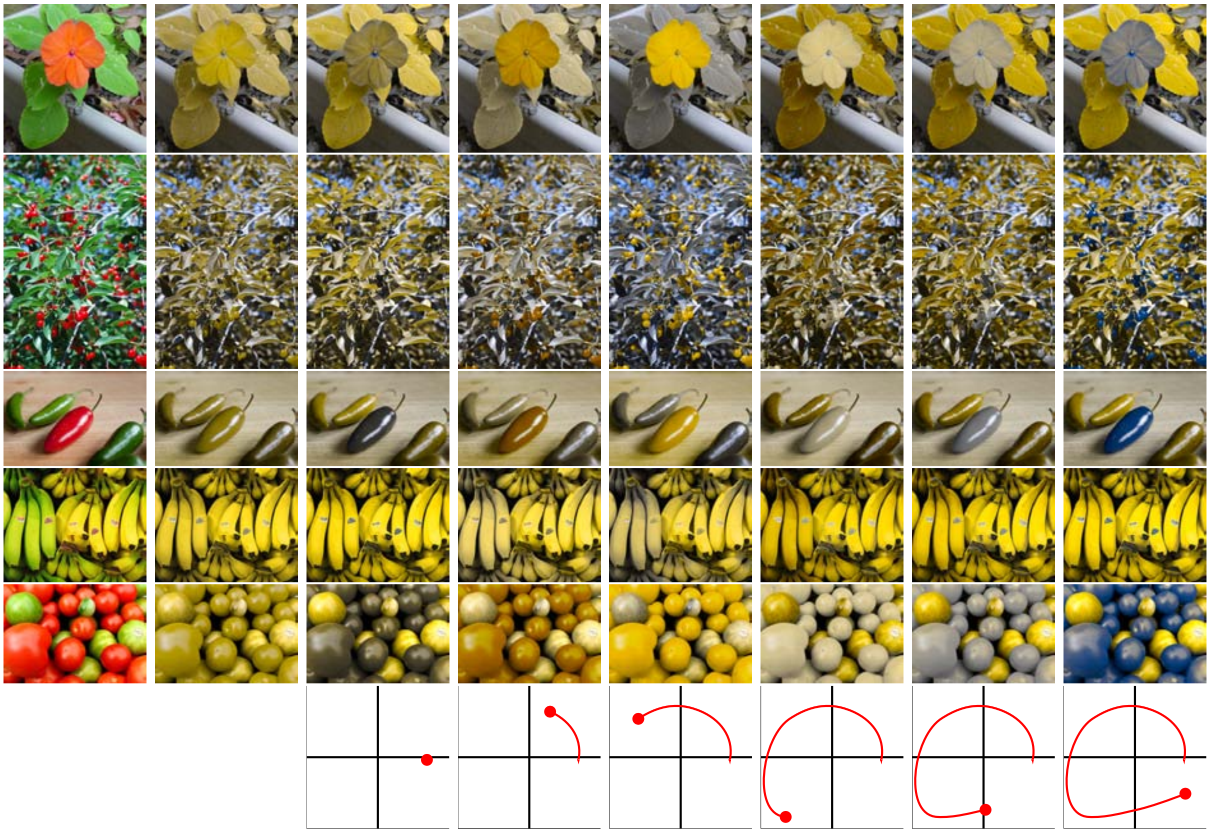


Figure 3. **Interactively change colors.** Our app allows viewers to interactively visualize color contrasts over time by transforming image colors according to the path the user traces on the touchscreen (last row). The first column contains the original images, and the second column simulates how a deuteranope would see the images. The rest of the columns simulate how deuteranopes would see the images transformed by the user input. Certain shears increase the contrast between red and green objects, allowing viewers to distinguish between them. *Original images courtesy of Rasche et.al,*[23] *Howard Walfish,*[24] *Pen Waggener,*[25] *Daniel Flck,*[26] *and Flickr user Mags_cat.*[27]

ΔE units. Although the CIELUV space is not perfect in its perceptual uniformity, a value of 2.9 approximates a just noticeable difference.[32] Thus, the color selection process ensures that the resulting set of six colors are clearly discriminable for a normal trichromatic observer, and the sets of three corresponding colors are clearly discriminable for an observer with the user's color deficiency. From each set of three colors along the observer's confusion line, a color is randomly chosen and replicated, resulting in the final set of eight colors. In a trial of the game, the user is presented with a montage of eight non-overlapping squares with the chosen colors. The task for the observer is to interact with the touchscreen and modify the colors until a correct match becomes evident to them, which is then selected by simultaneously touching two points in the screen lying in the corresponding squares. Additionally, a time constraint of two minutes is introduced for the total of trials that the user may attempt, with the objective to establish a reference for evaluation, and also to increase the appeal of the task.

As a control experiment, users are also asked to play the game with the method disabled. In this situation, if the user truly cannot distinguish the patches designed to be confusing for their deficiency type, random guessing would result in one correct choice out of $\binom{4}{2}$ possible pairs, i.e., a 17% accuracy in selection of the correct matches. As an additional validation control, we also asked normal trichromatic observers to play the game.

Figure 4 summarizes the results obtained from the game used for our validation. The interactive manipulation of the colors using the proposed method in the smartphone app significantly improves the accuracy of finding a true match. With the method disabled, four of the users had an accuracy under 23% and one user had an accuracy of 58%. On the other hand, with the method enabled, four of the users achieved an accuracy of 100%

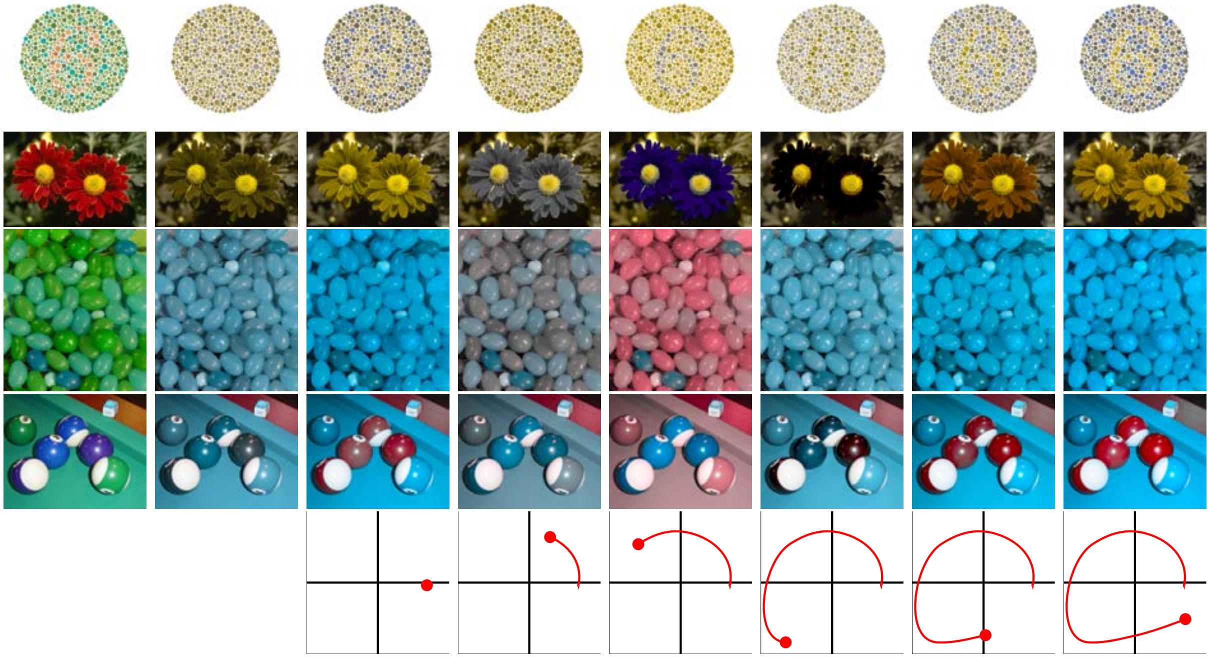


Figure 4. **Accommodates different types of color deficient viewers.** Our method can reveal hidden contrasts for protanopic (rows 1 and 2) and tritanopic (rows 3 and 4) viewers as well as deuteranopic viewers (Figure 3). The first column contains the original images, and the second column simulates how a color deficient viewer would see the images. The rest of the columns simulate how the color deficient viewer would see the images transformed by the user input. Through the different transformations, protanopic and tritanopic users can interactively visualize the red-green and blue-green contrast, respectively. *Original images courtesy of Kuhn et al.,*[28] *and Rasche et.al*[29]*.*[30]

whereas a fifth user did marginally worse with an accuracy of 83%. The four normal trichomats all achieved an accuracy of 100% validating the fact that the choice of the colors made the non-replicated colors clearly distinguishable.

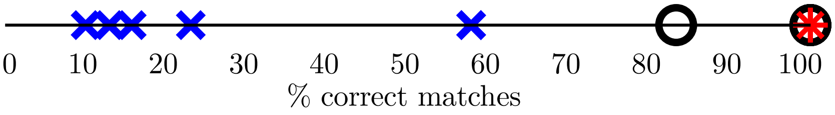


Figure 5. **Evaluation through game.** The percentage of correct matches obtained in the course of the game is higher for the five tested observers when interaction using our method is enabled (○) than when interaction is disabled (×). As a reference, the percentage of correct matches for a normal trichromatic observer is also indicated (∗). Repeats occur at the 100% accuracy four times for both ○ and ∗. The results clearly indicate that the interactive visualization using the proposed method in the app significantly improves the accuracy of finding the correct matches.

5. CONCLUSION

We present a smartphone app that helps color deficient viewers visualize contrasts they have difficulty seeing. With our app, the user can interactively change the colors in the image by tracing a path on the touchscreen. We transform the pixel colors based on a simple shear and translation operation that is performed efficiently on the GPU, allowing us to achieve real-time rates. Our method continuously and smoothly shears the data so that different information is revealed with each shear. As the viewer moves around the touchscreen, color contrasts change, revealing information from the third dimension of color that is present in individuals with normal color vision but shifted or missing for those with a color deficiency. This allows color deficient viewers to interactively visualize problematic color contrasts as changes in contrast over time.

Although the app we presented specifically targets interactive visualization for color deficient viewers, our methodology for interactively incorporating information is extensible to visualizing imagery with four channels. Specifically, the shearing transformation can be defined for 4D data to interactively incorporate information from the fourth channel into the three visibly rendered channels and thereby visualize the data in the fourth channel as changes in image colors over time. This could be used, for example, with common satellite multispectral images that include an infrared channel in addition to the traditional RGB channels. The proposed extension would allow the user to start with the visible image and interactively incorporate different amounts of information from the infrared image. As the user moves their finger over the touchscreen, they change the shearing transform parameters producing a change in the color contrasts in the displayed image. The real-time interaction would allow the user to rapidly explore different renderings.

REFERENCES

[1] "Color blindness," (2014). Accessed Feb 2014. http://en.wikipedia.org/wiki/Color_blindness.

[2] Flück, D., "Living with Color Blindness," (2010). http://www.color-blindness.com/2010/03/30/living-with-color-blindness/.

[3] Flatla, D. R. and Gutwin, C., ""So That's What You See!" Building Understanding with Personalized Simulations of Colour Vision Deficiency," in [*Proc. 14th Int. ACM SIGACCESS Conf. Computers and Accessibility*], 167–174, ACM (2012).

[4] Brettel, H., Viénot, F., and Mollon, J., "Computerized Simulation of Color Appearance For Dichromats," *J. Opt. Soc. Am. A* **14**(10), 2647–2655 (1997).

[5] zaveqna, "Apples," (2008). Accessed Feb 2014. https://www.flickr.com/photos/zaveqna/2872120203/.

[6] Dougherty, R. and Wade, A., "Daltonize." Accessed Sep 2014. http://www.vischeck.com/daltonize/.

[7] "Google Chrome Daltonize!." Accessed Sep 2014. https://chrome.google.com/webstore/detail/efeladnkafmoofnbagdbfaieabmejfcf.

[8] Wakita, K. and Shimamura, K., "SmartColor: Disambiguation Framework for the Colorblind," in [*Proc. ACM SIGACCESS Conf. Computers and Accessibility*], 158–165 (2005).

[9] Jefferson, L. and Harvey, R., "An Interface to Support Color Blind Computer Users," in [*Proc. SIGCHI Conf. Human Factors in Computing Systems*], 1535–1538 (2007).

[10] Machado, G. and Oliveira, M., "Real-Time Temporal-Coherent Color Contrast Enhancement For Dichromats," in [*Proc. 12th Eurographics/IEEE-VGTC Conf. Visualization*], 993–942 (2010).

[11] Flatla, D. and Gutwin, C., "SSMRecolor: Improving Recoloring Tools with Situation-Specific Models of Color Differentiation," in [*Proc. 30th Int. Conf. Human Factors in Computing Systems*], 2297–2306 (2012).

[12] Sajadi, B., Majumder, A., Oliveira, M., Schneider, R., and Raskar, R., "Using Patterns to Encode Color Information for Dichromats," *IEEE Trans. Visualization and Computer Graphics* **19**(1), 118–129 (2013).

[13] Rasche, K., Geist, R., and Westall, J., "Re-coloring Images for Gamuts of Lower Dimension," *Computer Graphics Forum/Eurographics* **24**(3), 423–432 (2005).

[14] Kuhn, G., Oliveira, M., and Fernandes, L., "An Efficient Naturalness-Preserving Image-Recoloring Method for Dichromats," *IEEE Trans. Visualization and Computer Graphics* **14**(6), 1747–1754 (2008).

[15] Eynard, D., Kovnatsky, A., and Bronstein, M., "Laplacian Colormaps: A Framework For Structure-Preserving Color Transformations," *Computer Graphics Forum/Eurographics* **33**(2), 215–224 (2014).

[16] Hung, P. and Hiramatsu, N., "A colour conversion method which allows colourblind and normal-vision people share documents with colour content," tech. rep., Konica Minolta Tech. Report (2013).

[17] Rodríguez-Pardo, C. E. and Sharma, G., "Adaptive color visualization for dichromats using a customized hierarchical palette," in [*Proc. SPIE: Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications*], Eschbach, R., Marcu, G. G., and Rizzi, A., eds., **7866**, 7866–03,1–9 (Jan. 2011).

[18] Flück, D., "20 iPhone Apps for the Color Blind," (2010). Accessed Sep 2014. http://www.colblindor.com/2010/12/13/20-iphone-apps-for-the-color-blind/.

[19] Kaminsky, D., "DanKam: Colorblind Fix," (2010). Accessed Sep 2014. http://itunes.apple.com/us/app/dankam-colorblind-fix/id406739331?mt=8.

[20] NGHS.fr, "Color blindness correction." Accessed Sep 2014. http://www.androidzoom.com/android_applications/medical/color-blindness-correction_bcqip.html.

[21] Deadly Apps, "Enliven - Color Blind Aid," (2010). Accessed Sep 2014. http://www.deadlyapps.com/2010/09/enliven-color-blind-aid.html.

[22] GamuProg, "Color Unblinder." Accessed Sep 2014. http://www.appszoom.com/android_applications/health/color-unblinder_bepvw.html.

[23] Rasche, K., Geist, R., and Westall, J., "Impatien," (2005). Accessed Feb 2014. http://people.cs.clemson.edu/~westall/recolor/recolor.html.

[24] Walfish, H., "Cherry picking," (2009). Accessed Feb 2014. https://www.flickr.com/photos/h-bomb/3787114217/.

[25] Waggener, P., "Peppers," (2011). Accessed Feb 2014. https://www.flickr.com/photos/epw/7537026064/.

[26] Flück, D., "Bananas," (2010). Accessed Feb 2014. http://www.color-blindness.com/2010/03/30/living-with-color-blindness/.

[27] Mags_cat, "Red and Green," (2007). Accessed Feb 2014. https://www.flickr.com/photos/mk1971/1304376620/in/photostream/.

[28] Kuhn, G., Oliveira, M., and Fernandes, L., "Red Flower," (2008). Accessed Feb 2014. http://www.inf.ufrgs.br/~oliveira/pubs_files/CVD_Mass_Spring/CVD_Mass_Spring.html.

[29] Rasche, K., Geist, R., and Westall, J., "JellyBeans," (2005). Accessed Feb 2014. http://people.cs.clemson.edu/~westall/recolor/recolor.html.

[30] Rasche, K., Geist, R., and Westall, J., "Balls," (2005). Accessed Feb 2014. http://people.cs.clemson.edu/~westall/recolor/recolor.html.

[31] Von Ahn, L. and Dabbish, L., "Designing Games With a Purpose," *Communications of the ACM* **51**(8), 58–67 (2008).

[32] Mahy, M., Eyckden, L. V., and Oosterlinck, A., "Evaluation of Uniform Color Spaces Developed After the adoption of CIELAB and CIELUV," *Color Res. Appl.* **19**, 105–121 (Apr. 1994).